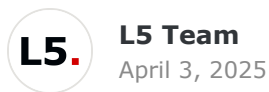


ClickUp and Make.com: How to Connect Them and What to Automate

Connect ClickUp to Make.com step by step, learn the exact modules to use, and build 8 no-code automations. Make vs Zapier vs ClickUp native, compared. Verified July 2026.



Last verified July 2026.

Connecting ClickUp to Make.com takes about five minutes and no code. You create a scenario in Make, add a ClickUp module, and authorize it with your ClickUp account over OAuth. From there Make can watch your workspace for a change, a new task, a status update, a moved subtask, and fire off a chain of actions across thousands of other apps. It is how you push ClickUp past the automations it can run on its own.

ClickUp's built-in automations are excellent inside a single Space or List. They start to run out of room the moment a workflow needs to reach another tool, transform data along the way, branch on a condition, or touch parts of the API the native builder does not expose. Make.com is the layer that fills that gap. This guide covers how to connect the two, the exact modules you get, eight automations worth building, and how Make stacks up against both Zapier and ClickUp's own automations, with current pricing and API limits.

The Short Version

Make.com extends ClickUp past its built-in automations. You connect the two over OAuth in about five minutes, then automate work across thousands of apps with no code.

- *The connection is OAuth. Add a ClickUp module in Make, click Create a connection, sign in to ClickUp, and grant access. No API key needed for the standard app.*
- *Make gives you roughly 90 ClickUp modules: triggers that watch tasks, lists, folders and spaces, actions that create and edit tasks and custom fields, and a catch-all Make an API call module for anything the prebuilt modules miss.*
- *Use ClickUp's native automations for single-Space rules. Reach for Make when a workflow crosses apps, needs branching or math, or has to update custom fields in bulk.*
- *Make bills by operation, not by task. Paid plans start at about \$9 per month. A free plan runs 1,000 operations a month on a 15-minute schedule.*
- *ClickUp's API allows 100 requests per minute per token on most plans. A busy scenario can hit that ceiling, so design for it.*

What the ClickUp and Make.com integration actually is

Make.com (formerly Integromat) is a visual, no-code automation platform. You build a **scenario**, a flow of connected modules, where the first module is usually a **trigger** (something happens) and the rest are **actions** (do this, then this). Make talks to ClickUp through ClickUp's public API, so a scenario can read and write almost anything in your workspace: tasks, subtasks, custom fields, comments, checklists, time tracking, goals, folders, and lists.

The difference that matters: ClickUp's own automations live inside ClickUp and act on ClickUp. Make sits outside and connects ClickUp to the rest of your stack, with logic in between. A native automation can move a task when its status changes. A Make scenario can watch that same status change, look up the customer in your CRM, calculate a total from the subtasks, draft an invoice in QuickBooks, and post a summary to Slack, in one run.

How to connect ClickUp to Make.com, step by step

You need a Make account (the free plan is enough to start) and a ClickUp account with permission to authorize integrations. The whole thing is done in Make.

1. Create a new scenario

In Make, open Scenarios and click Create a new scenario.

2. Add a ClickUp module

Click the plus, search for ClickUp, and pick your first module. If ClickUp starts the flow, choose a trigger such as *Watch Tasks*. If ClickUp is a later step, choose an action such as *Create a Task*.

3. Create a connection

In the module's Connection field, click Create a connection. Make uses OAuth, so a ClickUp window opens.

4. Authorize in ClickUp

Sign in if prompted, pick the Workspace you want Make to reach, and click Connect Workspace to grant access. You are returned to Make with a live connection you can reuse across every scenario.

5. Configure the module and test

Select the Team, Space, Folder, or List the module should act on, map your fields, then click Run once to fire a single test. When the module shows a green check and real data, the connection works.

When you need the advanced OAuth path

The standard connection covers almost everyone. If you are building for many workspaces or need a branded consent screen, Make also supports entering your own ClickUp OAuth client ID and secret under the connection's advanced settings, using the redirect URI Make provides. Most teams never touch this.

The ClickUp modules you get in Make

Make ships roughly 90 ClickUp modules, split across triggers, actions, and searches. You will use a handful constantly and reach for the rest occasionally.

Triggers (start a scenario)

- **Watch Tasks**, fires when a task or subtask is created or updated in a List you choose.
- **Watch Lists, Watch Folders, Watch Spaces**, fire when those containers change.
- **Instant triggers (webhooks)**, for modules with the *instant* tag, Make sets up a webhook so ClickUp pushes the event the moment it happens instead of Make polling on a schedule. Instant triggers are faster and cheaper on operations.

Actions (do something in ClickUp)

Module	What it does
Create a Task / Create a Task (advanced)	Adds a task. The advanced version exposes every field, including assignees, tags, priority, and dates.
Create a Task from a Template	Spins up a task, or a whole structure, from a saved ClickUp template.
Edit a Task with Custom Fields (advanced)	Updates a task and writes to custom fields by field ID. This is the workhorse for anything data-driven.
Get a Task / Delete a Task	Reads a single task's full payload, or removes it.
Add Tag , Add Tracked Time, Add a Dependency	Small, targeted edits to an existing task.
Create a Folder / List / Folderless List / Goal / Checklist	Builds the containers and checklists your workflow needs on the fly.
Make an API Call	The escape hatch. Calls any ClickUp API endpoint directly, so you are never blocked by a missing prebuilt module.

Searches (find records)

Search modules return lists of matching records to loop over, for example finding all tasks that match a filter, or listing the accessible custom fields in a List so you can grab the field IDs the *Edit a Task with Custom Fields* module needs.

8 ClickUp and Make.com automations to build

These are the scenarios teams reach for most. Each one names the trigger and the key actions so you can rebuild it.

1. Turn Slack messages into ClickUp tasks

Trigger on a Slack event (a message with a keyword, or a reaction like a checkmark), then *Create a Task* in the right List with the message text as the description and the sender mapped to an assignee. Requests stop living in threads and start living as trackable work.

2. Sync new ClickUp tasks to Google Sheets

Watch Tasks in ClickUp, then *Add a Row* in Google Sheets with the task name, status, assignee, and due date. Add a second scenario that watches the sheet and edits the task to make it a two-way sync. This is the simplest way to give finance or leadership a live view without buying them ClickUp seats.

3. Keep ClickUp and Notion in step

When a ClickUp task reaches a status like *Published*, create or update the matching Notion page, and vice versa. Teams that plan in ClickUp but publish docs in Notion use this to avoid double entry.

4. Roll subtask custom fields up to the parent

ClickUp cannot natively sum a number or money field from subtasks into the parent. In Make, search the subtasks, aggregate the field with a numeric aggregator, then use *Edit a Task with Custom Fields (advanced)* to write the total back to the parent. This gives you automatic budget, hours, or points rollups.

5. Draft a QuickBooks invoice when a task is completed

Watch Tasks for a status change to *Done* or *Approved*, pull the client and amount from custom fields, then create a draft invoice in QuickBooks (or Xero, or Stripe). Billing stops waiting on someone to remember it.

6. Spin up a full List from a template when you onboard a client

This was the original promise of pairing ClickUp with Make, and it still holds. Trigger on a new client (a form submission, a CRM record, or a new row), then *Create a Task from a Template* or create a List and populate it, so every new engagement starts from the same, consistent structure without manual copying.

7. Route and assign tasks across teams and Spaces

When marketing, sales, and support share a project across separate Spaces, a native automation cannot coordinate them. In Make, watch the source List, branch on a condition (department, priority, region), then assign the right owner, update a synced custom field, and move or mirror the task into the receiving Space.

8. Color-code Lists by status or risk with a webhook

Point an instant webhook trigger at events from a risk log or status change, then use *Make an API Call* to update a List's color. Green, amber, and red List colors update themselves, giving leadership a portfolio view that is always current instead of manually maintained.

A note on operations

Every module step in a scenario counts as one operation against your Make plan. A five-module scenario that runs 200 times a day is 1,000 operations a day. Favor instant (webhook) triggers over polling, filter early so scenarios exit before doing unnecessary work, and batch where you can. See the cost section below.

Make.com vs ClickUp's native automations

This is not an either/or choice. Most teams run both. The question is which tool owns a given workflow.

Question	ClickUp native automations	Make.com
Where it runs	Inside ClickUp	Outside, connecting ClickUp to other apps
Best for	Rules within one Space or List: status changes, assignments, due-date nudges	Cross-app flows, branching, data transformation, bulk custom-field updates
Other apps	A limited set of built-in integrations	Thousands of apps in one scenario
Logic	Trigger, condition, action	Multi-step scenarios with routers, filters, iterators, and aggregators
Pricing model	Included, capped by a monthly automation-runs limit per plan	Separate subscription, billed by operation

ClickUp's native automations are metered: plans run from 100 automation runs a month on Free up to hundreds of thousands on Enterprise, and the workspace pauses automations for the month once you hit the cap. For the full per-plan breakdown, see our [ClickUp pricing guide](#). The rule of

thumb: if a workflow never leaves ClickUp, build it natively and save your Make operations for the flows that cross tools.

Make.com vs Zapier for ClickUp

Both connect ClickUp to thousands of apps with no code. They differ in shape and in how they bill.

	Make.com	Zapier
Builder	Visual canvas, see the whole data path at once	Linear, step-by-step list
Complex logic	Strong: routers, filters, iterators, aggregators built in	Workable, but multi-path logic is clunkier
App library	Around 3,000 apps	Larger, roughly 8,000 apps
Billing	Per operation (each module step)	Per task (each completed action)
At high volume	Usually cheaper for multi-step flows	Costs climb faster on multi-step Zaps
Learning curve	Steeper, more power once learned	Gentler, faster for simple flows

Choose **Zapier** when the automation is simple, linear, and you value the largest app catalog and the fastest setup. Choose **Make** when a ClickUp workflow has real logic, several branches, data that needs reshaping, or high volume where per-operation billing wins. For ClickUp specifically, the custom-field and subtask-rollup work tends to be easier and cheaper in Make.

What it costs to run ClickUp and Make.com automations

Two meters matter: Make's operations, and ClickUp's API rate limit.

Make.com pricing

Make bills by operation. Every module step that runs is one operation. Plans, billed annually, are roughly:

Plan	Price (annual)	Operations / mo	Min. run interval
Free	\$0	1,000	15 minutes
Core	~\$9	10,000	1 minute
Pro	~\$16	10,000, plus priority execution and full logs	1 minute
Teams	~\$29	Team roles and shared templates on top of Pro	1 minute
Enterprise	Custom	Custom	1 minute

The free plan is enough to prototype and to run a couple of low-volume scenarios. The moment you want sub-15-minute reactions or more than 1,000 operations a month, you are on a paid plan. Prices and operation counts change, so confirm on [Make's pricing page](#) before you commit.

ClickUp API rate limits

Make talks to ClickUp through the API, which is capped per token:

ClickUp plan	Requests / minute / token
Free, Unlimited, Business	100
Business Plus	1,000
Enterprise	10,000

Go over the limit and the API returns an HTTP 429 and rejects the request. It also sends back X-RateLimit-Remaining and X-RateLimit-Reset headers so a well-built scenario can back off and retry. On the 100-per-minute plans, a scenario that loops over hundreds of tasks will hit the ceiling, so add delays, break work into batches, and lean on webhooks instead of frequent polling.

Common problems, and how to avoid them

- **Hitting the 429 rate limit**

Slow loops with a Sleep module, batch requests, and prefer instant webhook triggers over polling every minute. Handle the 429 with a break-and-retry error handler.

- **Custom fields will not update**

Custom fields are written by field ID, not by name. Run a *List Accessible Custom Fields* search once to grab the IDs, then map them into *Edit a Task with Custom Fields (advanced)*.

- **Operations burning faster than expected**

Each module step counts. A scenario that fans out over many records multiplies quickly. Filter early so the scenario exits before it does expensive work, and remove modules you added while testing.

- **Duplicate tasks from polling**

A polling trigger that re-reads the same records can double-create. Use the trigger's built-in de-duplication, or key on the ClickUp task ID before creating.

- **Scenario runs but nothing changes**

Almost always a mapping or scope error: the module is pointed at the wrong Space or List, or a required field is empty. Use Run once and inspect each module's input and output bundles.

ClickUp and Make.com automation FAQ

How do I connect ClickUp to Make.com?

Create a scenario in Make, add a ClickUp module, click Create a connection, and authorize it in the ClickUp window that opens over OAuth by selecting your Workspace and granting access. The connection is then reusable across every scenario. No API key is required for the standard app.

Is the ClickUp and Make.com integration free?

Yes, to start. Make's free plan includes 1,000 operations a month with a 15-minute minimum interval, and ClickUp's API is available on every plan. You move to a paid Make plan (from about \$9 a month) when you need faster reactions or more operations.

Do I need to know how to code?

No. Scenarios are built by dragging modules onto a canvas and mapping fields between them. The one power-user module, *Make an API call*, lets a technical user reach any ClickUp endpoint, but you can build the large majority of useful automations without it.

Make.com or Zapier for ClickUp: which is better?

Zapier is better for simple, linear automations and has a larger app catalog. Make is better for ClickUp workflows with branching, data transformation, or bulk custom-field updates, and its per-operation billing is usually cheaper for multi-step flows. Many teams start on Zapier and move complex ClickUp work to Make.

When should I use Make instead of ClickUp's native automations?

Use native automations for rules that stay inside one Space or List. Use Make when a workflow crosses into other apps, needs conditional logic or math, updates custom fields in bulk, or touches parts of the ClickUp API the native builder does not expose.

Can Make update ClickUp custom fields?

Yes. The *Edit a Task with Custom Fields (advanced)* action writes to custom fields by their field ID. Use a search module to list the accessible custom fields and copy the IDs, then map your values in.

What are ClickUp's API rate limits, and will Make hit them?

ClickUp allows 100 requests per minute per token on Free, Unlimited, and Business, 1,000 on Business Plus, and 10,000 on Enterprise. A busy scenario that loops over many tasks can hit the 100-per-minute ceiling and get an HTTP 429, so add delays, batch requests, and prefer webhook triggers over frequent polling.

Why is Make usually cheaper than Zapier for ClickUp automations?

Make bills per operation (each module step), while Zapier bills per task (each completed action). For multi-step ClickUp workflows the per-operation model tends to cost less at volume, which is why data-heavy automations often land in Make.