mixpanel

MCP · PROMPT LIBRARY

Copy, paste, analyze.

Prompts for the full Mixpanel MCP tool surface: analytics queries, flows, session replay, dashboard creation, Lexicon management, and cross-tool workflows.

Tool annotations show what fires under the hood. Works with Claude, ChatGPT, Gemini, Cursor, Notion, and any MCP-compatible client.

Updated April 2026 · docs.mixpanel.com/docs/mcp

01 ORIENT

Orient

Map the project's schema before running any query. Skip this step and your analysis will break in quiet ways — wrong event names, bad filters, missing properties.

Find your projects

"What Mixpanel projects do I have access to? List them with their IDs."

Tools `Get-Projects`

Start here if you work across multiple projects. The project ID sets the scope for everything that follows.

Map schema to a business question

"I want to understand [onboarding completion / feature adoption / checkout conversion] in this project. Look at available events and properties and suggest the top 3–5 candidates that best represent that behavior."

Tools `Get-Events, then Get-Event-Details on each candidate`

Quality depends on your Lexicon descriptions. Sparse descriptions produce vague suggestions.
Follow up: "Give me full details on [Event Name], including any data quality issues."

Understand what a specific event actually means

"Give me full details on [Event Name]: its description, what properties are attached, when it fires, and any data quality issues."

Tools `Get-Event-Details, Get-Issues (filtered to that event), Get-Property-Names`

Find your activation signal

"Based on the events in this project, what's the best candidate for an activation event — something that predicts whether a new user will stick around? Explain why each candidate does or doesn't fit."

Tools `Get-Events, Get-Event-Details on multiple candidates`

Reasoning task, not a query task. You need a retention query afterward to validate the candidate it identifies.

Explore properties and their actual values

"What properties are available on [Event Name]? For the ones useful as filters or breakdowns, show me the actual values each one contains."

Tools `Get-Property-Names, then Get-Property-Values for each relevant property`

Knowing actual values prevents bad filters. Run this before any breakdown prompt.

Get a direct link to Lexicon

"Give me the Lexicon URL for [Event Name] so I can edit it in the Mixpanel UI."

Tools `Get-Lexicon-URL`

02 ANALYZE

Analyze

Specify four things in every prompt: the behavior (which events), the population (who), the timeframe (when), and the shape of the answer (rate, trend, breakdown). Missing any one forces the AI to guess.

FUNNELS

Conversion funnel with a specific population

> "What's the conversion rate from [Step 1] to [Step 2] to [Step 3] for [first-time users / paid accounts / mobile users] over the last 30 days? Show me where the biggest drop-off is."

Tools `Get-Query-Schema (funnels)`, `Run-Query (funnels)`

Be explicit about population and timeframe. Run Orient prompts first if you're unsure which property to filter on.
Follow up: "Break that down by [plan type / device / channel]."

Week-over-week funnel trend

> "Show me how conversion from [Event A] to [Event B] has changed week over week for the past 8 weeks. Flag any weeks where it dropped more than 10%."

Tools `Get-Query-Schema (funnels)`, `Run-Query (funnels, weekly unit)`

Flagging is AI reasoning over the data — not a separate tool call.

Funnel breakdown by segment

> "Break down [Event A] to [Event B] conversion by [plan type / channel / device] for the last 60 days. Which segment converts best, which converts worst, and what's the gap?"

Tools `Get-Query-Schema (funnels)`, `Run-Query (funnels, with breakdown)`

The AI needs the exact property name. Add: "First confirm what properties are available for this event."

RETENTION

N-day retention from a starting event

> "What's the 1, 7, 14, and 30-day retention for users who completed [Starting Event] in the last 90 days? Use [Return Event] as the retention signal."

Tools `Get-Query-Schema (retention)`, `Run-Query (retention)`

Follow up: "Compare that to users who completed [Alternative Starting Event]."

Retention comparison across two time periods

> "Compare 30-day retention for users who signed up in [Month 1] vs [Month 2]. Did retention improve? Which part of the curve changed?"

Tools `Get-Query-Schema (retention)`, `Run-Query ×2`

Filters by signup date in the query, not a saved segment. Runs two queries and compares.

FEATURE ADOPTION

Adoption rate for a specific feature

> "What percentage of [active users / paid accounts / users who completed onboarding] have used [Feature Event] at least once in the last 30 days? How does that compare to the prior 30 days?"

Tools `Get-Query-Schema (insights)`, `Run-Query (insights, unique users) ×2`

Feature usage depth

> "For users who've used [Feature Event] at least once, how many times do they use it per week on average? Break it down by [plan type / user role] if those properties exist."

Tools `Get-Property-Names (confirm breakdown)`, `Get-Query-Schema (insights)`, `Run-Query (insights, with breakdown)`

Verify the breakdown property exists first — a missing property fails silently.

TRENDS

Event volume trend

> "Show me daily volume of [Event Name] over the last 30 days. Flag any days with a spike or drop greater than 20% from the 7-day average."

Tools `Get-Query-Schema (insights)`, `Run-Query (insights, daily)`

Pull a saved report

> "Retrieve the report named [Report Name] and show me the current results."

Tools `Get-Report (with results)`

Use Search-Entities to find the report by name if you don't know the exact title.

Comparative period analysis

> "Compare this month to last month across: [DAU], [Signup-to-Purchase conversion], and [7-day retention]. Summarize what improved, what got worse, and what stayed flat."

Tools `Run-Query (insights) ×2`, `Run-Query (funnels) ×2`, `Run-Query (retention) ×2`

Six-query session. Avoid breakdowns in the comparative pass — add them as a follow-up.
Follow up: "Break that funnel down by [channel / device] to see which segment drove the change."

FLOWS

Flows discover the most frequent routes users take to or from any event. Unlike funnels, you don't need to know the steps in advance.

What do users do after a key event?

> "Show me the 3 most common steps users take after [Event Name] in the last 30 days. Where do they go and how many drop off at each step?"

Tools `Get-Query-Schema (flows)`, `Run-Query (flows, stepsAfter=3, chartType=sankey)`

> On one e-commerce project: 75% of users who viewed their cart immediately removed a product — invisible in a standard funnel.

What leads to a conversion event?

"What are the 3 most common paths users take before [Purchase Completed / Subscription Started / Activation Event] in the last 30 days?"

Tools `Get-Query-Schema (flows), Run-Query (flows, stepsBefore=3, chartType=sankey)`

Use sankey chart type for stepsBefore — the paths chart type may return empty results.

Compare paths between two segments

"Compare the top paths after [Onboarding Completed] for [free users] vs [paid users] in the last 30 days. Where do their journeys diverge?"

Tools `Run-Query (flows with filter) ×2`

Discover what happens between two funnel steps

"My funnel from [Step A] to [Step B] has low conversion. Show me the most common paths users take between those two events."

Tools `Get-Query-Schema (flows), Run-Query (flows, Step A as starting event, stepsAfter=3)`

FEATURE LAUNCH EVALUATION

End-of-sprint feature assessment

"We shipped [Feature Name] two weeks ago. Run in order: 1) weekly adoption trend for the last 4 weeks, 2) funnel from first use to repeat use, 3) retention for users who adopted vs. didn't, 4) top paths after [Feature Event]. Summarize whether the launch is on track."

Tools `Run-Query (insights), Run-Query (funnels), Run-Query (retention), Run-Query (flows) — 4+ calls`

4–7 tool call session. The AI plans queries, runs them sequentially, and synthesizes.
Follow up: "Pull session replays for 5 users who tried the feature but didn't come back."

Why: Adoption numbers alone lie. A feature can show growing unique users but zero repeat usage and terrible retention. All four report types together give you the full picture.

Investigate

Zoom in on specific users, accounts, and sessions. These prompts move from aggregate data to individual behavior. Pair with session replay to see the experience firsthand.

Replay for a specific user

> "Pull session replays for user [distinct_id] from the last 14 days. Show what they were doing around the time they [dropped off / triggered an error]."

Tools `Get-User-Replays-Data (distinct_id + date range)`

Most reliable replay prompt. Works best when you already have a user ID from a prior analysis or support ticket.

Replay after identifying users via analysis (two-step)

> "Step 1: Show me the IDs of users who reached [Step N] but didn't complete [Final Event] in the last 7 days. Return up to 10. Step 2: Pull session replays for those users."

Tools `Step 1: Run-Query (funnels, user-level). Step 2: Get-User-Replays-Data`

MCP returns replay metadata and links. You open recordings in the Mixpanel UI.

Diagnose a drop-off user

> "User [distinct_id] was active last month but hasn't logged in for 2 weeks. Pull their last 30 days of activity and replays from their final sessions."

Tools `Run-Query (insights, filtered to user), Get-User-Replays-Data`

Review a user's recent sessions

> "Pull the last 5 sessions for user [distinct_id]. For each, show features used, session length, and where they exited. Are sessions getting shorter over time?"

Tools `Get-User-Replays-Data, Run-Query (insights, filtered to user, daily)`

Follow up: "Show me the replay for their shortest session."

Account-level investigation

> "Show me all activity for users at [Account Name / company_id] over the last 30 days. Which features are they using most? Where are they dropping off?"

Tools `Get-Events, Run-Query (insights, filtered to account)`

Requires account-level properties on events (e.g., company_id). Without those, you're working with user-level data only.

Follow up: "Pull session replays for their most active user."

Build

Create and manage dashboards, and audit what exists in your project. Dashboards created via MCP are real Mixpanel dashboards that persist and are visible to your teammates.

DASHBOARDS

Create a dashboard from scratch

> "Create a Mixpanel dashboard called [Dashboard Name] with: 1) daily active users over 30 days, 2) signup-to-purchase funnel, 3) 7-day retention for new users. Add a text card at the top summarizing what this board tracks."

Tools `Create-Dashboard`

Build a weekly growth dashboard

> "Build a dashboard tracking signups, activations, and churn week over week for 12 weeks. Include a text card explaining the metric definitions."

Tools `Create-Dashboard`

Run Orient prompts first if you're unsure about the event names for signup, activation, or churn.

Duplicate and customize an existing dashboard

> "Duplicate the [Dashboard Name] dashboard, change the date range to last quarter, and rename it [New Dashboard Name]."

Tools `Get-Dashboard, Duplicate-Dashboard`

Update a dashboard's layout

> "In [Dashboard Name], move the retention report to the top row and add a text card below it explaining how we define retention."

Tools `Get-Dashboard (include_layout=True), Update-Dashboard`

Call Get-Dashboard with include_layout=True first to get cell and row IDs.

Build a dashboard from your analysis

> "Take the three queries we just ran and create a Mixpanel dashboard called [Dashboard Name]. Add a text card at the top summarizing the findings."

Tools `Create-Dashboard (using query specs from prior conversation context)`

AUDIT PROJECT ARTIFACTS

List and audit existing dashboards

> "What dashboards exist in this project? Which ones haven't been updated in 90 days?"

Tools `Search-Entities (entity_types=['dashboard'], sort_by='last_edited')`

Full project artifact inventory

> "Give me an inventory of everything: dashboards, reports, cohorts, experiments, feature flags, metric trees. For each type, tell me the count and when the most recent was last edited."

Tools `Search-Entities (with each entity type)`

> Running this on one project surfaced feature flags untouched since December 2025 and dozens of unnamed metric trees.

05 GOVERN

Govern

Lexicon management and data quality triage. These are write operations — requires Project Owner or Admin role. Useful at scale where doing this manually in the UI takes hours.

Find stale experiments and feature flags

> "List all experiments and feature flags. Which haven't been edited in 60 days? Who created them?"

Tools `Search-Entities (entity_types=['experiments', 'feature-flags'], sort_by='last_edited')`

LEXICON MANAGEMENT

Add descriptions to undocumented events

> "Find all events without a Lexicon description. For each one, suggest a description based on the event name and properties. Then apply the descriptions."

Tools `Get-Events, Get-Event-Details (loop), Edit-Event (loop)`

Review suggestions before confirming — the AI can guess wrong on internal shorthand.

Tag related events

> "Tag all events related to [checkout / onboarding / search] with the tag "[Tag Name]". Create the tag if it doesn't exist."

Tools `Get-Events, Create-Tag (if needed), Edit-Event (loop)`

Hide inactive events

> "Find all events that haven't fired in the last 90 days and hide them in Lexicon."

Tools `Get-Events (with volume data), Edit-Event (set hidden, loop)`

Hidden events are still queryable by name — this just cleans up the browsing UI.

Flag PII properties

> "Look at all properties across events. Flag any that might contain PII (email, phone, name, IP) but aren't marked sensitive yet."

Tools `Get-Events, Get-Property-Names (loop), Get-Property-Values (sample), Edit-Property (set sensitive)`

Catches obvious cases. May miss obfuscated or custom fields — treat as a starting audit.

Rename a tag across the project

> "Rename the tag "[Old Tag]" to "[New Tag]" across all events and properties."

Tools `Rename-Tag`

DATA QUALITY

Audit data quality for a specific event

> "Are there open data quality issues for [Event Name]? Summarize what's broken, when each was detected, and which to fix first."

Tools `Get-Issues (filtered by event_name)`

Prioritization is AI judgment, not a severity score. Treat the ranking as a starting point.

Full project health check

"Run a data quality audit across all events and properties. Group issues by severity and tell me which are most likely affecting analysis right now."

Tools `Get-Issues (no filters)`

Check data quality before critical analysis

"Before I run a funnel on [Event A] through [Event C], check for any data quality issues on those three events."

Tools `Get-Issues (filtered to each event)`

Follow up: "If clean, run the funnel. If not, fix the issues first."

Escalate stale issues to engineering

"Find all data quality issues open more than 14 days. Format the list so I can paste it into a message to the data engineering team."

Tools `Get-Issues (no filters)`

Chain

Combine MCP data with attached files, external context, or other connected tools. Run analysis prompts first, then use these to pull it together.

COMBINE WITH ATTACHED FILES

Correlate a metric change with external events

> "Our [metric] changed between [date range]. I'm attaching our [campaign calendar / release notes]. Does the timing correlate with anything in this document?"

Tools `No MCP tools – AI reasons across prior query data and the attached file`

Attach the file alongside this prompt. Needs prior query results in the same conversation.

QBR prep with template

> "I'm preparing a QBR for [Account Name]. I've attached our slide template. Pull their usage data for last quarter and fill in the template with adoption rates, engagement trends, and areas of concern."

Tools `Run-Query (insights, filtered to account), then AI fills template from attached file`

COMBINE WITH OTHER CONNECTED TOOLS

If your AI client has multiple MCP servers connected, you can chain Mixpanel data with other tools in a single conversation.

Mixpanel + error monitoring (Sentry, Datadog)

> "Pull users who triggered [Error Event] more than 3 times this week. Then check our error monitoring tool for matching exceptions. Same root cause or different bugs?"

Tools `Run-Query (Mixpanel MCP), then tool calls to the error monitoring MCP`

Mixpanel + team messaging (Slack, Teams)

> "Pull adoption numbers for [Feature X] from Mixpanel, then search [#customer-feedback] for mentions of [Feature X]. Summarize the quantitative and qualitative signals together."

Tools `Run-Query (Mixpanel MCP), then search/read calls to the messaging MCP`

> **Why:** High adoption with negative sentiment is a different problem than low adoption with no mentions at all.

Mixpanel + project planning (Linear, Jira, Asana)

> "Show me the top 3 drop-off points in [Core Funnel]. Then check our project tracker for open issues at those points — known bugs or feature requests that might explain the drops?"

Tools `Run-Query (Mixpanel MCP, funnels), then search calls to the project planning MCP`

Mixpanel + calendar (Google Calendar, Outlook)

> "Check my calendar for external meetings this week. For each company, pull their usage data from Mixpanel for the last 30 days. Summarize in 3 bullets with anything that changed week-over-week."

Tools `Calendar MCP (list events), then Run-Query (Mixpanel MCP) per meeting`

Requires a company/account property on events. The AI loops through meetings and queries per account.

Principles

How to get reliable results — and how to avoid failure modes that produce confident-looking wrong answers.

Specify four things in every analysis prompt

The behavior (which events), the population (who), the timeframe (when), and the shape of the answer (rate, trend, breakdown). Leave any out and the AI fills in defaults that may not match what you wanted.

Start with schema discovery

Before running funnels or retention queries, ask MCP which events represent the behavior you care about. It reads your actual project schema — you don't have to guess event names.

Check property names and values before breakdowns

Add "first confirm what properties are available on this event" to avoid silent failures. Checking actual values avoids filters that match nothing.

Use Flows for discovery, Funnels for measurement

Funnels measure the rate between steps you already know. Flows discover the steps you don't. When conversion is low and you're not sure why, Flows shows what users are doing instead. Use sankey chart type for stepsBefore queries.

MCP cannot access saved segments or cohorts

Express population filters as event or user properties. "Users where plan_type = enterprise" works. "My Enterprise cohort" does not.

Follow up in the same conversation

MCP retains context within a session. "Break that down by plan type" or "compare that to last month" works without restating everything.

Check data quality before critical analysis

A broken event produces confident-looking wrong answers. Ask about data quality issues first on any event you haven't used recently.

Results don't save back to Mixpanel (except dashboards)

Query results in a conversation are temporary. Recreate the report in the Mixpanel UI or use Create-Dashboard to persist it. Dashboards created via MCP are real, persistent Mixpanel dashboards.

Search-Entities finds more than reports and dashboards

It also discovers experiments, feature flags, metric trees, playlists, heat maps, and cohorts — useful for auditing project hygiene. Drill into full details with Get-Dashboard or Get-Report.

Cross-tool workflows require multiple MCP connections

To chain Mixpanel data with Slack, Jira, Notion, or Sentry, those MCP servers must be connected in the same AI client and authorized.

Write operations need Project Owner or Admin role

Editing events, properties, tags, and dashboards requires the right permissions. If a write operation fails silently, check your role first.

Rate limit: 600 requests per hour

Heavy analysis sessions can approach this limit. If calls start failing, wait and retry. For 20+ query sessions, break them into multiple sessions.

REFERENCE

Tool Reference

All 25 MCP tools as of April 2026. See docs.mixpanel.com/docs/mcp for current documentation.

CATEGORY	TOOL	WHAT IT DOES
ANALYTICS		
	`Run-Query`	Insights, funnels, flows, retention queries
	`Get-Query-Schema`	JSON schema for building a query
	`Get-Report`	Retrieve a saved report with optional results
DASHBOARDS		
	`Create-Dashboard`	New dashboard with text cards and reports
	`List-Dashboards`	Browse dashboards in a project
	`Get-Dashboard`	Dashboard metadata, text cards, reports
	`Update-Dashboard`	Modify metadata, rows, layout
	`Duplicate-Dashboard`	Copy an existing dashboard
	`Delete-Dashboard`	Delete a dashboard
DISCOVERY		
	`Get-Projects`	List projects and workspaces
	`Get-Events`	Browse all events
	`Get-Property-Names`	Properties for events or users
	`Get-Property-Values`	Values for a specific property
	`Get-Event-Details`	Full event metadata
	`Get-Issues`	Data quality issues
	`Get-Lexicon-URL`	Direct link to Lexicon entry
	`Search-Entities`	Search dashboards, reports, cohorts, experiments, feature flags, metric trees, playlists, heat maps
MANAGEMENT		
	`Get-Property`	Full metadata for a single property
	`Edit-Event`	Update description, tags, visibility
	`Edit-Property`	Update description, hidden, PII flag
	`Create-Tag`	New Lexicon tag
	`Rename-Tag`	Rename tag across all associations
	`Delete-Tag`	Remove a tag
	`Dismiss-Issues`	Bulk-dismiss data quality issues
REPLAYS		
	`Get-User-Replays-Data`	Session replays for a specific user