



unitQ monitor MCP Server

Connect your AI assistant to your unitQ customer feedback and query it in plain language. This guide works with any MCP-capable assistant, with Claude, ChatGPT, and Microsoft Copilot used as examples.

SERVER URL	<code>https://mcp.unitq.com/v1/mcp</code>
TRANSPORT	Streamable HTTP (remote, hosted)
AUTHENTICATION	OAuth 2.0 with PKCE, through your organization's identity provider

Overview

unitQ monitor uses AI to analyze customer feedback across every channel in real time. It centralizes signals from support tickets, app store reviews, social, surveys, and more, then classifies them with a custom multi-tier taxonomy.

The unitQ monitor MCP server lets an MCP-capable AI assistant, such as Claude, ChatGPT, or Microsoft Copilot, query that feedback for you. In a single conversation you can:

- Find what customers are complaining about, with the actual quotes and deep links back into unitQ.
- Count and break down feedback by category, platform, source, app version, sentiment, and more.
- Surface AI-clustered themes and subtopics that the fixed taxonomy may not cover.
- Track metrics like unitQ Score, Net Sentiment, and Average Rating over time.
- Read your existing dashboards, and build or update dashboards directly from the conversation.

Every request runs against your own unitQ data, scoped to what your user account is already permitted to see.

Prerequisites

Before connecting, you need:

1. **A unitQ account** on a company (tenant) with feedback data.

2. **MCP access enabled for your user.** Access is off by default. A unitQ admin turns it on per user under **Manage Users > Permissions > System Access > MCP server toggle**. If the connector signs in but no tools appear, this is almost always the reason.

No API keys, personal access tokens, or pasted secrets are required at any point.

Connecting the server

Any assistant that supports remote MCP servers can connect to unitQ monitor. The steps below describe the general flow. Exact menu labels and plan availability differ by product, so treat your assistant's own connector settings and documentation as the source of truth.

1. Open your assistant's **connector or integrations settings** and choose to add a custom or remote MCP server.
2. Paste the server URL: `https://mcp.unitq.com/v1/mcp`
3. Start the connection. The assistant opens your organization's sign-in.
4. **Sign in through your identity provider** (SSO), completing multi-factor authentication if prompted.
5. **Approve the consent screen** to authorize the assistant to access your unitQ data on your behalf.
6. Confirm the unitQ monitor tools now appear in the assistant.

Where to find the setting in common assistants:

- **Claude** (web and desktop): Settings > Connectors > Add custom connector.
- **ChatGPT**: the connector or app settings used to add a custom MCP server, on plans that support custom connectors.
- **Microsoft Copilot**: the area where custom or remote MCP tools are added, on plans that support them.

Tokens refresh automatically once connected, so you should not need to sign in again for normal use.

To confirm it is working, ask something like: "What are the top complaint categories for our iOS app this week?"

Tool reference

The server exposes 19 tools: 17 read tools and 2 write tools. Read tools are annotated read-only. Both write tools act only on dashboards and never modify or delete feedback.

Read tools

TOOL	WHAT IT DOES
Search Feedback (<code>query_feedback</code>)	Search feedback across every channel using semantic understanding or exact keyword matching. Returns the actual items with quotes and deep links back into unitQ.
Count Feedback (<code>count_feedback</code>)	Return a fast total count of feedback matching a query or filter combination, without pulling the items.
Analyze Feedback by Category (<code>query_feedback_categories</code>)	Analyze feedback by the category taxonomy with volume, period-over-period change, timeseries, and breakdowns by dimension.
Get Category Details (<code>get_feedback_categories</code>)	Fetch full metadata for specific categories, including description, status, and position in the taxonomy.
Discover Subtopics (<code>query_feedback_subtopics</code>)	Discover emergent themes with AI clustering, with proportions, estimated counts, and example quotes.
Get Subtopic Feedback IDs (<code>get_subtopic_feedback_ids</code>)	Return the feedback IDs behind a discovered subtopic so a follow-up can be scoped to that theme.
List Filters (<code>list_filters</code>)	List the filter dimensions available for a tenant, such as platform, source, and sentiment.
Query Filter Values (<code>query_filters</code>)	Show the values within a filter dimension and how feedback is distributed across them.
List Metrics (<code>list_metrics</code>)	List the metrics available for a tenant, such as unitQ Score, Net Sentiment, and Average Rating.
Query Metric (<code>query_metrics</code>)	Query a single metric over time, with optional timeseries and breakdowns.
Query Alerts (<code>query_alerts</code>)	Query live or historical alerts by status, returning critical, warning, and ok states.
Search Charts (<code>query_charts</code>)	Search and list the reusable charts a tenant has across all dashboards.
Search Dashboards (<code>query_dashboards</code>)	Search and list the dashboards available to a tenant.
Get Dashboard Data (<code>get_dashboard_data</code>)	

TOOL	WHAT IT DOES
	Fetch a single dashboard by ID with the full data for every chart on it.
Get Dashboard Composition (<code>get_dashboard_composition</code>)	Fetch a dashboard's structure: its layout, widgets, notes, and chart arrangement.
Dashboard Layout Reference (<code>describe_dashboard_layout</code>)	Return the reference for how dashboard layouts work, including grid dimensions and geometry.
Chart Config Reference (<code>describe_chart_config</code>)	Return a minimal working example of a chart configuration for a given chart type.

Write tools

TOOL	WHAT IT DOES
Create Dashboard (<code>create_dashboard</code>)	Create a new dashboard for a tenant, optionally with an initial set of charts. Additive; it does not modify existing data.
Update Dashboard (<code>save_dashboard_composition</code>)	Replace or update an existing dashboard's composition. This is the only tool that overwrites existing dashboard structure.

Before running either write tool, your assistant should present the proposed layout and wait for your explicit confirmation. `save_dashboard_composition` is annotated as destructive because it overwrites the target dashboard's structure, so review the proposed changes before approving.

Example prompts

Find what customers are saying, with quotes - "What are users saying about onboarding? Show me representative quotes." - "What are the biggest drivers of negative reviews right now?"

Quantify and break down feedback - "Count feedback mentioning login problems on Android in the last 30 days." - "Break down negative feedback by app version for the current quarter." - "Which platform, iOS or Android, has worse sentiment in the last 30 days?"

Track categories, subtopics, and metrics over time - "What are the top complaint categories for our iOS app this week?" - "Which subtopics are driving negative feedback this month?" - "How has Net Sentiment trended over the last quarter?"

Build dashboards from the conversation - “Build a dashboard tracking checkout complaints by platform.” - “Create a dashboard for weekly sentiment across our top five categories.”

Security and access

ASPECT	DETAIL
Authentication	OAuth 2.0 with PKCE through your identity provider, with user consent and MFA.
Credentials	No API keys or pasted tokens. No long-lived credentials in the client. Tokens refresh automatically.
Access default	Off by default. Enabled per user by a unitQ admin through the MCP server toggle.
Data scoping	Scoped to the authenticated company. One customer cannot see another’s data.
Per-user permissions	A user only sees the data they are already permitted to see in unitQ.
Third-party access	The server calls only unitQ first-party APIs on a unitQ domain. No open-web access and no third-party data retrieval.
Data handling	Accesses only data explicitly requested by the user. No data is stored beyond session requirements. All transmission is encrypted over HTTPS/TLS.

Troubleshooting

The connector signs in, but no tools or data appear. MCP access is likely not enabled for your user. Ask a unitQ admin to turn on the MCP server toggle for your account (**Manage Users > Permissions > System Access**), then reconnect.

Sign-in loops or authorization fails. Make sure you complete the full identity-provider flow, including MFA. If it still fails, remove the connector and add it again so a fresh authorization runs. Confirm you are signing in with your organization’s SSO account.

Queries return no results. Confirm you are scoped to the correct company (tenant) and that feedback exists in the time window you asked about. Queries default to the last 30 days unless you specify a range, so widen the window if a recent spike is expected but nothing appears.

A filter or filter value is rejected. Filter dimensions and their values are specific to each tenant. Use “list the available filters” and “show the values for [dimension]” so your assistant can discover the exact names and values your account supports, rather than guessing.

Category numbers look different from a keyword search. Analyzing a named category and searching for that word as free text are different operations. Ask your assistant to analyze the category by name so it scopes to feedback assigned to that category, rather than counting every item that merely mentions the word.

A dashboard change did not save. Both write tools require your explicit confirmation in chat before they run. If you did not approve the proposed layout, no change is written. `save_dashboard_composition` overwrites the existing dashboard structure, so review the proposal before confirming.

The server URL shows nothing in a browser. This is expected. The endpoint is not a web page; it responds only to the MCP client handshake and OAuth flow. Connect it as a connector rather than opening the URL directly.

Support and resources

- **Support:** support@unitq.com
- **Privacy policy:** <https://www.unitq.com/privacy-policy/>
- **Terms of use:** <https://www.unitq.com/terms-of-use/>
- **Product overview:** <https://www.unitq.com/products/monitorq/>

For access requests or permission changes, contact your unitQ administrator.