Model Context Protocol

Get Started › Examples

Get Started

Examples

A list of example servers and implementations

This page showcases various Model Context Protocol (MCP) servers that demonstrate the protocol's capabilities and versatility. These servers enable Large Language Models (LLMs) to securely access tools and data sources.

Reference implementations

These official reference servers demonstrate core MCP features and SDK usage:

Data and file systems

Filesystem - Secure file operations with configurable access controls

PostgreSQL - Read-only database access with schema inspection capabilities

SQLite - Database interaction and business intelligence features

Google Drive - File access and search capabilities for Google Drive

Development tools

Git - Tools to read, search, and manipulate Git repositories

GitHub - Repository management, file operations, and GitHub API integration

GitLab - GitLab API integration enabling project management

Sentry - Retrieving and analyzing issues from Sentry.io

Web and browser automation

Model Context Protocol

- **Brave Search** - Web and local search using Brave's Search API
- **Fetch** - Web content fetching and conversion optimized for LLM usage
- **Puppeteer** - Browser automation and web scraping capabilities

Productivity and communication

- **Slack** - Channel management and messaging capabilities
- **Google Maps** - Location services, directions, and place details
- **Memory** - Knowledge graph-based persistent memory system

AI and specialized tools

- **EverArt** - AI image generation using various models
- **Sequential Thinking** - Dynamic problem-solving through thought sequences
- **AWS KB Retrieval** - Retrieval from AWS Knowledge Base using Bedrock Agent Runtime

Official integrations

These MCP servers are maintained by companies for their platforms:

- **Axiom** - Query and analyze logs, traces, and event data using natural language
- **Browserbase** - Automate browser interactions in the cloud
- **Cloudflare** - Deploy and manage resources on the Cloudflare developer platform
- **E2B** - Execute code in secure cloud sandboxes
- **Neon** - Interact with the Neon serverless Postgres platform
- **Obsidian Markdown Notes** - Read and search through Markdown notes in Obsidian vaults
- **Qdrant** - Implement semantic memory using the Qdrant vector search engine
- **Raygun** - Access crash reporting and monitoring data

Search1API - Unified API for search, crawling, and sitemaps
Model Context Protocol
Tinybird - Interface with the Tinybird serverless ClickHouse platform

Community highlights

A growing ecosystem of community-developed servers extends MCP's capabilities:

Docker - Manage containers, images, volumes, and networks

Kubernetes - Manage pods, deployments, and services

Linear - Project management and issue tracking

Snowflake - Interact with Snowflake databases

Spotify - Control Spotify playback and manage playlists

Todoist - Task management integration

> **Note:** Community servers are untested and should be used at your own risk. They are not affiliated with or endorsed by Anthropic.

For a complete list of community servers, visit the **MCP Servers Repository**.

Getting started

Using reference servers

TypeScript-based servers can be used directly with `npx`:

```
npx -y @modelcontextprotocol/server-memory
```

Python-based servers can be used with `uvx` (recommended) or `pip`:

```
# Using uvx
uvx mcp-server-git
```

```
# Using pip
pip install mcp-server-git
python -m mcp_server_git
```


Configuring with Claude

To use an MCP server with Claude, add it to your configuration:

```
{
  "mcpServers": {
    "memory": {
      "command": "npx",
      "args": ["-y", "@modelcontextprotocol/server-memory"]
    },
    "filesystem": {
      "command": "npx",
      "args": ["-y", "@modelcontextprotocol/server-filesystem", "/path/to/allowed
    },
    "github": {
      "command": "npx",
      "args": ["-y", "@modelcontextprotocol/server-github"],
      "env": {
        "GITHUB_PERSONAL_ACCESS_TOKEN": "<YOUR_TOKEN>"
      }
    }
  }
}
```

Additional resources

MCP Servers Repository - Complete collection of reference implementations and community servers

Awesome MCP Servers - Curated list of MCP servers

MCP CLI - Command-line inspector for testing MCP servers

‹ **Quickstart** **Clients** ›