≡ **Model Context Protocol**

≡ **Get Started** › **Clients**

Get Started

Clients

A list of applications that support MCP integrations

This page provides an overview of applications that support the Model Context Protocol (MCP). Each client may support different MCP features, allowing for varying levels of integration with MCP servers.

Feature support matrix

Client	Resources	Prompts	Tools	Sampling	Roots	Notes
Claude Desktop App	✅	✅	✅	❌	❌	Full support for all MCP features
Zed	❌	✅	❌	❌	❌	Prompts appear as slash commands
Sourcegraph Cody	✅	❌	❌	❌	❌	Supports resources through OpenCTX
Firebase Genkit	⚠️	✅	✅	❌	❌	Supports resource list and lookup through tools.
Continue	✅	✅	✅	❌	❌	Full support for all MCP features
GenAIScript	❌	❌	✅	❌	❌	Supports tools.
Cline	✅	❌	✅	❌	❌	Supports tools and resources.
LibreChat	❌	❌	✅	❌	❌	Supports tools for Agents

Client details

Claude Desktop App
Model Context Protocol

The Claude desktop application provides comprehensive support for MCP, enabling deep integration with local tools and data sources.

Key features:

Full support for resources, allowing attachment of local files and data

Support for prompt templates

Tool integration for executing commands and scripts

Local server connections for enhanced privacy and security

> ⓘ Note: The Claude.ai web application does not currently support MCP. MCP features are only available in the desktop application.

Zed

Zed is a high-performance code editor with built-in MCP support, focusing on prompt templates and tool integration.

Key features:

Prompt templates surface as slash commands in the editor

Tool integration for enhanced coding workflows

Tight integration with editor features and workspace context

Does not support MCP resources

Sourcegraph Cody

Cody is Sourcegraph's AI coding assistant, which implements MCP through OpenCTX.

Key features:

Support for MCP resources

Integration with Sourcegraph's code intelligence

Uses OpenCTX as an abstraction layer

Model Context Protocol

Future support planned for additional MCP features

Get Started › **Clients**

Firebase Genkit

Genkit is Firebase's SDK for building and integrating GenAI features into applications. The **genkitx-mcp** plugin enables consuming MCP servers as a client or creating MCP servers from Genkit tools and prompts.

Key features:

  Client support for tools and prompts (resources partially supported)

  Rich discovery with support in Genkit's Dev UI playground

  Seamless interoperability with Genkit's existing tools and prompts

  Works across a wide variety of GenAI models from top providers

Continue

Continue is an open-source AI code assistant, with built-in support for all MCP features.

Key features

  Type "@" to mention MCP resources

  Prompt templates surface as slash commands

  Use both built-in and MCP tools directly in chat

  Supports VS Code and JetBrains IDEs, with any LLM

GenAIScript

Programmatically assemble prompts for LLMs using **GenAIScript** (in JavaScript). Orchestrate LLMs, tools, and data in JavaScript.

Key features:

Cline

Cline is an autonomous coding agent in VS Code that edits files, runs commands, uses a browser, and more–with your permission at each step.

Key features:

Create and add tools through natural language (e.g. "add a tool that searches the web")

Share custom MCP servers Cline creates with others via the `~/Documents/Cline/MCP` directory

Displays configured MCP servers along with their tools, resources, and any error logs

LibreChat

LibreChat is an open-source, customizable AI chat UI that supports multiple AI providers, now including MCP integration.

Key features:

Extend current tool ecosystem, including **Code Interpreter** and Image generation tools, through MCP servers

Add tools to customizable **Agents**, using a variety of LLMs from top providers

Open-source and self-hostable, with secure multi-user support

Future roadmap includes expanded MCP feature support

Adding MCP support to your application

If you've added MCP support to your application, we encourage you to submit a pull request to add it to this list. MCP integration can provide your users with powerful contextual AI

capabilities and make your application part of the growing MCP ecosystem.

🖉 **Model Context Protocol**

Benefits of adding MCP support:

Enable users to bring their own context and tools

Join a growing ecosystem of interoperable AI applications

Provide users with flexible integration options

Support local-first AI workflows

To get started with implementing MCP in your application, check out our **Python** or **TypeScript SDK Documentation**

Updates and corrections

This list is maintained by the community. If you notice any inaccuracies or would like to update information about MCP support in your application, please submit a pull request or **open an issue in our documentation repository**.

Was this page helpful?   👍 Yes   👎 No

‹ **Examples**   **Building MCP clients** ›