Get Started

Example 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
5ire	❌	❌	✅	❌	❌	Supports tools.
BeeAI Framework	❌	❌	✅	❌	❌	Supports tools in agentic workflows.
Cline	✅	❌	✅	❌	❌	Supports tools and resources.
Continue	✅	✅	✅	❌	❌	Full support for all MCP features
Cursor	❌	❌	✅	❌	❌	Supports tools.
Emacs Mcp	❌	❌	✅	❌	❌	Supports tools in Emacs.
Firebase Genkit	⚠️	✅	✅	❌	❌	Supports resource list and lookup through tools.
GenAIScript	❌	❌	✅	❌	❌	Supports tools.
Goose	❌	❌	✅	❌	❌	Supports tools.
LibreChat	❌	❌	✅	❌	❌	Supports tools for Agents
mcp-agent	❌	❌	✅	⚠️	❌	Supports tools, server connection management, and agent workflows.
oterm	❌	❌	✅	❌	❌	Supports tools.
Roo Code	✅	❌	✅	❌	❌	Supports tools and resources.
Sourcegraph Cody	✅	❌	❌	❌	❌	Supports resources through OpenCTX
Superinterface	❌	❌	✅	❌	❌	Supports tools
TheiaAI/TheiaIDE	❌	❌	✅	❌	❌	Supports tools for Agents in Theia AI and the AI-powered Theia IDE

Client	Resources	Prompts	Tools	Sampling	Roots	Notes
Windsurf Editor	❌	❌	✅	❌	❌	Supports tools with AI Flow for collaborative development.
Zed	❌	✅	❌	❌	❌	Prompts appear as slash commands
SpinAI	❌	❌	✅	❌	❌	Supports tools for Typescript AI Agents
OpenSumi	❌	❌	✅	❌	❌	Supports tools in OpenSumi
Daydreams Agents	✅	✅	✅	❌	❌	Support for drop in Servers to Daydreams agents

Client details

Claude Desktop App

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.

5ire

5ire is an open source cross-platform desktop AI assistant that supports tools through MCP servers.

Key features:

- Built-in MCP servers can be quickly enabled and disabled.
- Users can add more servers by modifying the configuration file.
- It is open-source and user-friendly, suitable for beginners.
- Future support for MCP will be continuously improved.

BeeAI Framework

BeeAI Framework is an open-source framework for building, deploying, and serving powerful agentic workflows at scale. The framework includes the **MCP Tool**, a native feature that simplifies the integration of MCP servers into agentic workflows.

Key features:

- Seamlessly incorporate MCP tools into agentic workflows.
- Quickly instantiate framework-native tools from connected MCP client(s).
- Planned future support for agentic MCP capabilities.

Learn more:

- **Example of using MCP tools in agentic workflow**

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

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

Cursor

Cursor is an AI code editor.

Key Features:

- Support for MCP tools in Cursor Composer
- Support for both STDIO and SSE

Emacs Mcp

Emacs Mcp is an Emacs client designed to interface with MCP servers, enabling seamless connections and interactions. It provides MCP tool invocation support for AI plugins like **gptel** and **llm**, adhering to Emacs’ standard tool invocation format. This integration enhances the functionality of AI tools within the Emacs ecosystem.

Key features:

- Provides MCP tool support for Emacs.

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

GenAIScript

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

Key features:

- JavaScript toolbox to work with prompts

- Abstraction to make it easy and productive
- Seamless Visual Studio Code integration

Goose

Goose is an open source AI agent that supercharges your software development by automating coding tasks.

Key features:

- Expose MCP functionality to Goose through tools.
- MCPs can be installed directly via the **extensions directory**, CLI, or UI.
- Goose allows you to extend its functionality by **building your own MCP servers**.
- Includes built-in tools for development, web scraping, automation, memory, and integrations with JetBrains and Google Drive.

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

mcp-agent

mcp-agent is a simple, composable framework to build agents using Model Context Protocol.

Key features:

- Automatic connection management of MCP servers.
- Expose tools from multiple servers to an LLM.
- Implements every pattern defined in **Building Effective Agents**.
- Supports workflow pause/resume signals, such as waiting for human feedback.

oterm

oterm is a terminal client for Ollama allowing users to create chats/agents.

Key features:

- Support for multiple fully customizable chat sessions with Ollama connected with tools.
- Support for MCP tools.

Roo Code

Roo Code enables AI coding assistance via MCP.

Key features:

- Support for MCP tools and resources
- Integration with development workflows
- Extensible AI capabilities

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
- Future support planned for additional MCP features

SpinAI

SpinAI is an open-source TypeScript framework for building observable AI agents. The framework provides native MCP compatibility, allowing agents to seamlessly integrate with MCP servers and tools.

Key features:

- Built-in MCP compatibility for AI agents
- Open-source TypeScript framework
- Observable agent architecture
- Native support for MCP tools integration

Superinterface

Superinterface is AI infrastructure and a developer platform to build in-app AI assistants with support for MCP, interactive components, client-side function calling and more.

Key features:

- Use tools from MCP servers in assistants embedded via React components or script tags
- SSE transport support
- Use any AI model from any AI provider (OpenAI, Anthropic, Ollama, others)

TheiaAI/TheiaIDE

Theia AI is a framework for building AI-enhanced tools and IDEs. The **AI-powered Theia IDE** is an open and flexible development environment built on Theia AI.

Key features:

- **Tool Integration**: Theia AI enables AI agents, including those in the Theia IDE, to utilize MCP servers for seamless tool interaction.
- **Customizable Prompts**: The Theia IDE allows users to define and adapt prompts, dynamically integrating MCP servers for tailored workflows.
- **Custom agents**: The Theia IDE supports creating custom agents that leverage MCP capabilities, enabling users to design dedicated workflows on the fly.

Theia AI and Theia IDE’s MCP integration provide users with flexibility, making them powerful platforms for exploring and adapting MCP.

Learn more:

- **Theia IDE and Theia AI MCP Announcement**
- **Download the AI-powered Theia IDE**

Windsurf Editor

Windsurf Editor is an agentic IDE that combines AI assistance with developer workflows. It features an innovative AI Flow system that enables both collaborative and independent AI interactions while maintaining developer control.

Key features:

- Revolutionary AI Flow paradigm for human-AI collaboration
- Intelligent code generation and understanding
- Rich development tools with multi-model support

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

OpenSumi

OpenSumi is a framework helps you quickly build AI Native IDE products.

Key features:

- Supports MCP tools in OpenSumi
- Supports built-in IDE MCP servers and custom MCP servers

Daydreams

Daydreams is a generative agent framework for executing anything onchain

Key features:

- Supports MCP Servers in config
- Exposes MCP Client

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.

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

‹ Example Servers

Building MCP with LLMs ›