Get Started

Introduction

Get started with the Model Context Protocol (MCP)

Java SDK released! Check out **what else is new.**

MCP is an open protocol that standardizes how applications provide context to LLMs. Think of MCP like a USB-C port for AI applications. Just as USB-C provides a standardized way to connect your devices to various peripherals and accessories, MCP provides a standardized way to connect AI models to different data sources and tools.

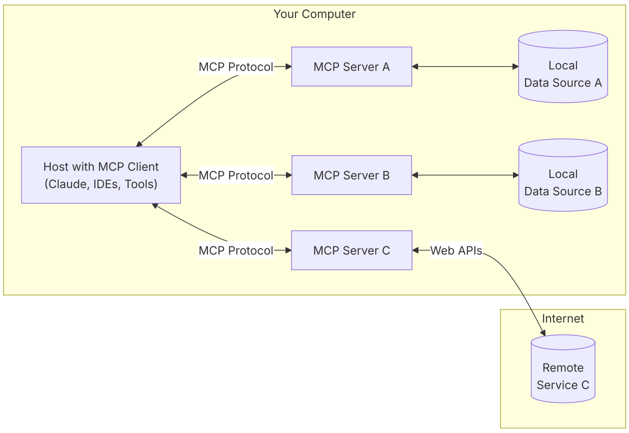
Why MCP?

MCP helps you build agents and complex workflows on top of LLMs. LLMs frequently need to integrate with data and tools, and MCP provides:

- A growing list of pre-built integrations that your LLM can directly plug into
- The flexibility to switch between LLM providers and vendors
- Best practices for securing your data within your infrastructure

General architecture

At its core, MCP follows a client-server architecture where a host application can connect to multiple servers:



- **MCP Hosts**: Programs like Claude Desktop, IDEs, or AI tools that want to access data through MCP
- **MCP Clients**: Protocol clients that maintain 1:1 connections with servers
- **MCP Servers**: Lightweight programs that each expose specific capabilities through the standardized Model Context Protocol
- **Local Data Sources**: Your computer’s files, databases, and services that MCP servers can securely access
- **Remote Services**: External systems available over the internet (e.g., through APIs) that MCP servers can connect to

Get started

Choose the path that best fits your needs:

Quick Starts

server to use in Claude for Desktop and other clients

client that can integrate with all MCP servers

For Claude Desktop Users

Get started using pre-built servers in Claude for Desktop

Examples

Example Servers

Check out our gallery of official MCP servers and implementations

Example Clients

View the list of clients that support MCP integrations

Tutorials

Building MCP with LLMs

Learn how to use LLMs like Claude to speed up your MCP development

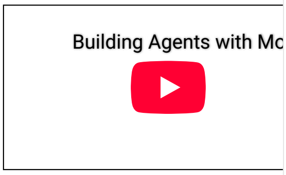
Debugging Guide

Learn how to effectively debug MCP servers and integrations

MCP Inspector

Test and inspect your MCP servers with our interactive debugging tool

MCP Workshop (Video, 2hr)



Explore MCP

Dive deeper into MCP’s core concepts and capabilities:

Core architecture

Understand how MCP connects clients, servers, and LLMs

Resources

Expose data and content from your servers to LLMs

Prompts

Create reusable prompt templates and workflows

Tools

Enable LLMs to perform actions through your server

Sampling

Let your servers request completions from LLMs

Transports

Learn about MCP’s communication mechanism

Support and Feedback

Here’s how to get help or provide feedback:

- For bug reports and feature requests related to the MCP specification, SDKs, or documentation (open source), please **create a GitHub issue**
- For discussions or Q&A about the MCP specification, use the **specification discussions**
- For discussions or Q&A about other MCP open source components, use the **organization discussions**
- For bug reports, feature requests, and questions related to Claude.app and claude.ai’s MCP integration, please email **mcp-support@anthropic.com**

Was this page helpful? 👍 Yes 👎 No

For Server Developers ›