Model Context Protocol (MCP)

Connecting AI Models with External Tools

Complete Introduction Guide

What is MCP?

💡 Simple idea: Connect AI models with external tools

Real-world needs:

- File access and manipulation
- Database connections and queries
- API integrations
- External service communications

Client ↔ **Host** ↔ **Server**

MCP Server

Definition

Server provides tools & resources to clients through the MCP protocol

Example: Weather Server

Weather server → provides current weather data, forecasts, and historical weather information

How servers work:

- Registers available tools with the host
- Responds to tool execution requests
- Manages resource access and data

MCP Server - Code Example

FastMCP

Search the docs... Ctrl K

jlowin/fastmcp 17,731

Documentation What's New SDK Reference

Get Started

- Welcome!
- Installation
- Quickstart

Servers

- Overview
- Core Components
- Advanced Features
- Authentication
- Deployment

Clients

- Essentials
- Core Operations

Welcome to FastMCP 2.0!

The fast, Pythonic way to build MCP servers and clients.

The **Model Context Protocol** (MCP) is a new, standardized way to provide context and tools to your LLMs, and FastMCP makes building MCP servers and clients simple and intuitive. Create tools, expose resources, define prompts, and more with clean, Pythonic code:

```python
from fastmcp import FastMCP

mcp = FastMCP("Demo 🚀")

@mcp.tool
def add(a: int, b: int) -> int:
    """Add two numbers"""
    return a + b

if __name__ == "__main__":
    mcp.run()
```

On this page

MCP Client

Definition

Client = application or AI model that uses the server's capabilities

Example: Claude Desktop

Claude Desktop acts as an MCP client, connecting to various MCP servers to expand Claude's capabilities

Demo: Client calling a server tool

1. Client discovers available tools from server
2. Client sends tool execution request
3. Server processes request and returns result
4. Client receives and uses the result

Analogy: Client = "user", Server = "service provider"

MCP Host

Definition

Host = middle layer between client and server that manages connections and communication

Host responsibilities:

- Manages multiple server connections
- Routes requests between clients and servers
- Handles protocol translation and validation
- Provides security and error handling

Example: Claude Desktop Host

Claude Desktop's host can connect to multiple MCP servers simultaneously (weather, database, file system, etc.)

Client → **Host** → **Server**

MCP Specifications

What are MCP specifications?

Standardized rules and protocols that define how MCP components communicate

Built on JSON-RPC Protocol

Uses JSON-RPC 2.0 for reliable, structured communication between components. JSON-RPC messages **MUST** be UTF-8 encoded.

Common specification parts:

- **Tools** - Define available actions and their parameters
- **Resources** - Specify data and file access methods
- **Prompts** - Standardize reusable AI instructions

Why standards matter

Standard rules = Easy integration

Any MCP-compliant client can work with any MCP-compliant server, regardless of who built them

Transport Mechanisms

Protocol Revision: 2025-06-18

MCP uses JSON-RPC to encode messages. JSON-RPC messages **MUST** be UTF-8 encoded.

Two Standard Transport Mechanisms:

1. stdio Transport

Communication over standard in and standard out

- Local process communication
- Direct stdin/stdout messaging
- Ideal for local servers
- Simple implementation

2. Streamable HTTP Transport

HTTP-based communication with streaming support

- Remote server connections
- Real-time bidirectional streaming
- Web-compatible transport
- Scalable for enterprise use

MCP Transport

Choosing the Right Transport

stdio: Best for local development and simple integrations

Streamable HTTP: Best for production, remote servers, and real-time applications

stdio Use Cases

- Local file system access
- Development and testing
- Simple command-line tools
- Single-user applications

HTTP Use Cases

- Cloud-based services
- Multi-user environments
- Real-time data streaming
- Enterprise deployments

✨ Both transports support the same MCP protocol features

You can switch between transports without changing your server logic!

MCP Tools, Prompts & Resources

Tools

Perform actions

- Search operations
- Database queries
- API calls
- File operations

Resources

Share data/files

- CSV files
- PDF documents
- Images
- Configuration data

Prompts

Reusable AI instructions

- Code review templates
- Analysis frameworks
- Writing guidelines
- Task instructions

📚 Official Documentation

- **MCP Specification:**
  modelcontextprotocol.io
- **Anthropic Docs:**
  docs.anthropic.com/en/docs/mcp
- **Getting Started:**
  modelcontextprotocol.io/quickstart

🛠️ SDKs & Tools

- **Python SDK:**
  `pip install mcp`
- **TypeScript SDK:**
  `npm install @modelcontextprotocol/sdk`
- **C# SDK (NuGet):**
  `ModelContextProtocol`

🐙 GitHub Resources

- **Official MCP Servers:**
  github.com/modelcontextprotocol/servers
- **Awesome MCP Servers:**
  github.com/punkpeye/awesome-mcp-servers

🌟 Popular Implementations

- **GitHub MCP Server:**
  github.com/github/github-mcp-server
- **Claude Desktop:**
  claude.ai/download