

Tutorials

Building MCP with LLMs

Speed up your MCP development using LLMs such as Claude!

This guide will help you use LLMs to help you build custom Model Context Protocol (MCP) servers and clients. We'll be focusing on Claude for this tutorial, but you can do this with any frontier LLM.

Preparing the documentation

Before starting, gather the necessary documentation to help Claude understand MCP:

1. Visit <https://modelcontextprotocol.io/llms-full.txt> and copy the full documentation text
2. Navigate to either the **MCP TypeScript SDK** or **Python SDK repository**
3. Copy the README files and other relevant documentation
4. Paste these documents into your conversation with Claude

Describing your server

Once you've provided the documentation, clearly describe to Claude what kind of server you want to build. Be specific about:

What resources your server will expose

What tools it will provide



Build an MCP server that:

- Connects to my company's PostgreSQL database
- Exposes table schemas as resources
- Provides tools for running read-only SQL queries
- Includes prompts for common data analysis tasks

Working with Claude

When working with Claude on MCP servers:

1. Start with the core functionality first, then iterate to add more features
2. Ask Claude to explain any parts of the code you don't understand
3. Request modifications or improvements as needed
4. Have Claude help you test the server and handle edge cases

Claude can help implement all the key MCP features:

Resource management and exposure

Tool definitions and implementations

Prompt templates and handlers

Error handling and logging

Connection and transport setup

Best practices

When building MCP servers with Claude:



Model Context Protocol

Break down complex servers into smaller pieces

Test each component thoroughly before moving on

Tutorials

Keep security in mind, validate inputs and limit access appropriately

Building MCP with LLMs

Document your code well for future maintenance

Follow MCP protocol specifications carefully

Next steps

After Claude helps you build your server:

1. Review the generated code carefully
2. Test the server with the MCP Inspector tool
3. Connect it to Claude.app or other MCP clients
4. Iterate based on real usage and feedback

Remember that Claude can help you modify and improve your server as requirements change over time.

Need more guidance? Just ask Claude specific questions about implementing MCP features or troubleshooting issues that arise.

Was this page helpful?



Yes



No

[Example Clients](#)

[Debugging](#)

