



The Context Layer · Following

 Member-only story

How I Use CodeMCP For Development And Keep My Other Coding Assistants On Budget

5 min read · Jun 3, 2025



Jannis

[Follow](#) Listen Share More

While tools like Cursor and Windsurf offer incredible capabilities, their token-based pricing can quickly add up during intensive coding sessions. Here's how I strategically use CodeMCP to handle specific development tasks before reaching for my paid assistants.

Welcome back. You are signed into your member account
bg****@jaxondigital.com.



Why CodeMCP First?

The core advantage of free open source CodeMCP is that it's designed to work with Claude Pro's flat \$20/month subscription, helping developers "say goodbye to giant API bills". This fixed-cost model is perfect for the exploratory and iterative work that often burns through tokens in other tools.

Getting Started: The 15-Minute Setup

The setup process is surprisingly straightforward, and I've streamlined it into a workflow that takes about 15 minutes from start to finish. Here's exactly how I got CodeMCP running on my machine:

Step 1: Install Prerequisites (5 minutes) First, you'll need `uv` (Python's fast package installer) and `git` if they're not already installed:

- **Install uv:** Visit docs.astral.sh/uv/getting-started/installation and follow the one-line install for your system
- **Install git:** If you don't have it, grab it from git-scm.com (Windows users: reboot after installation)

Step 2: Configure Claude Desktop (5 minutes)

Next, you'll need to edit Claude Desktop's configuration file. It will look something like this:

```
Welcome back. You are signed into your member account  
bg****@jaxondigital.com.
```

- Windows:

```
C:\Users\<username>\AppData\Roaming\Claude\claude_desktop_config.json
```

- macOS:

```
~/Library/Application Support/Claude/claude_desktop_config.json
```

Open this file in any text editor and add:

```
{  
  "mcpServers": {  
    "codemcp": {  
      "command": "/Users/<username>/.local/bin/uvx",  
      "args": [  
        "--from",  
        "git+https://github.com/ezyang/codemcp@prod",  
        "codemcp"  
      ]  
    }  
  }  
}
```

Windows users: Use double backslashes in the path:

```
C:\\\\Users\\\\<username>\\\\.local\\\\bin\\\\uvx.exe
```

Step 3: Test the Installation (5 minutes) Restart Claude Desktop completely. If everything worked, you'll see a hammer icon (🔨) appear in the interface. Click it and verify “codemcp” appears in the tools list.

If something goes wrong, check the MCP logs at Settings > Developer > codemcp > Logs. The error messages there are usually very helpful for troubleshooting.

My CodeMCP-First Workflow

1. Project Setup and Initial Exploration

When I start working on a new codebase, I use CodeMCP for the initial reconnaissance. The tool provides comprehensive file system operations including reading files, listing directories, and searching through code with grep. This exploration phase would typically consume significant tokens in other tools, but

with CodeMCP, I can freely browse and understand the codebase structure without worrying about token usage.

Welcome back. You are signed into your member account

bg.....@jaxondigital.com.

2. Code Formatting and Basic Maintenance

CodeMCP excels at routine maintenance tasks. I configure formatting commands in my `codemcp.toml` file and let Claude handle code formatting across entire projects. These repetitive tasks are perfect for CodeMCP since they don't require the advanced reasoning I might need from premium tools.

3. Targeted File Edits and Refactoring

The EditFile tool allows for precise, targeted modifications. I use this for straightforward refactoring tasks like renaming variables, updating function signatures, or making consistent changes across multiple files. CodeMCP handles these mechanical changes reliably while preserving my token budget for more complex reasoning tasks.

4. Git Operations and Version Control

One of CodeMCP's strongest features is its integrated git workflow. Every change generates a commit automatically, and the tool provides git blame, diff, log, and show capabilities. I use this for code archaeology, understanding change history, and maintaining clean commit histories without burning tokens on git-related queries in other tools.

5. Test Running and Basic Debugging

By configuring test commands in my project setup `zshrc`, I can have CodeMCP run tests and iterate on fixes for failing test cases. This is particularly valuable for TDD workflows where I'm making small, incremental changes.

When I Switch to Premium Tools

After CodeMCP handles the groundwork, I switch to Cursor or Windsurf for:

- Complex architectural decisions that require deep reasoning
- Multi-file refactoring that needs sophisticated understanding of code relationships
- Advanced debugging of complex logic issues
- Feature planning and design discussions

The Safety Net Advantage

CodeMCP's emphasis on "safe agentic AI" with Git version control integration means I can "unleash" my AI to handle more complex tasks. Welcome back. You are signed into your member account [REDACTED] long, everything [REDACTED] be more aggressive with automation for routine tasks.

My Budget Impact

This workflow has reduced my monthly spending on token-based coding assistants by approximately 60%. Tasks that would have cost \$50–100 in tokens monthly now happen within my Claude Pro subscription. I reserve my premium tool budgets for the high-value, complex reasoning tasks where they can shine.

Project Configuration Strategy

I maintain a standard `codemcp.toml` template that I drop into new projects within 5 minutes. Here's my battle-tested template and the reasoning behind each section:

```
project_prompt = """
You are working on [PROJECT_NAME]. Before making any changes:
1. Review the existing code structure
2. Follow the established patterns and naming conventions
3. Run tests after significant changes
4. Commit changes with descriptive messages
"""
```

```
[commands]
format = ["npm run format"] # or ["black .", "isort ."] for Python
test = ["npm test"] # or ["pytest"] for Python
lint = ["npm run lint"] # or ["flake8 ."] for Python
build = ["npm run build"] # or ["python setup.py build"] for Python
```

My project onboarding checklist:

1. Copy the template into the project root as `codemcp.toml`
2. Customize the commands to match your project's scripts
3. Set up the Claude Desktop project with the instruction: Initialize codemcp with `/path/to/your/project`
4. Test with a simple task like "format all files" to ensure everything works

Real-world example: For my React projects, I use:

Welcome back. You are signed into your member account
bg****@jaxondigital.com.

```
[commands]
format = ["prettier --write .", "eslint --fix ."]
test = ["npm test -- --watchAll=false"]
build = ["npm run build"]
type-check = ["npx tsc --noEmit"]
```

This setup pays dividends immediately. Instead of manually running these commands or burning tokens asking other AI tools to format code, I simply tell Claude “format the codebase” and it handles everything.

Notes

CodeMCP is particularly effective for developers who work on multiple projects or do significant code maintenance work. The flat-rate pricing model through Claude Pro makes it ideal for exploratory work, code cleanup, and routine development tasks. However, it requires some setup and configuration per project, and works best when integrated into a Git-based workflow. The tool is designed to be IDE-agnostic, so it complements rather than replaces your existing development environment.

[GitHub - ezyang/codemcp: Coding assistant MCP for Claude Desktop](#)

Coding assistant MCP for Claude Desktop. Contribute to ezyang/codemcp development by creating an account on GitHub.

[github.com](https://github.com/ezyang/codemcp)

[ezyang/codemcp](#)

Coding assistant MCP for Claude Desktop

36 Issues

1k Stars

112 Forks

If you found this article helpful, please consider giving it a clap .

And if you hold that button down something magically will happen, Try it!

Don't forget to follow me to stay updated on my latest posts. Together, we can continue to explore fascinating topics and expand our knowledge.

Thank you for your time and engagement!

Claude

Developer Tools

Mcp Server

Mcps

AI

Welcome back. You are signed into your member account
bg****@jaxondigital.com.

Following

Published in The Context Layer

720 followers · Last published 1 day ago

A journal of real-world experiments, tools, and thinking around Anthropic's Model Context Protocol (MCP)—now established as an open standard for large language model (LLM) context integration.



Follow



Written by Jannis



2.6K followers · 5 following

Product Owner in global telecom, lifelong tech tinkerer, and Mac user. Sharing hands-on hacks, real stories, and the tools that make work (and life) smarter.

Responses (3)



Bgerby

What are your thoughts?



Maksym Holovanov

Jun 24

...

MCP can help with some small typical part of code or some small setup as idea to go next only. Do you know your code and algorithms? How support this project next after year for example

[Reply](#)

Welcome back. You are signed into your member account
bg**@jaxondigital.com.**



Chris C
Sep 4

...

I don't understand how this will save your premium model usage if it uses Claude Pro or Max anyway - unless you configure it with a smaller model ?

[1 reply](#)[Reply](#)

Benjamin Sanders
Jun 20

...

Honestly curious. Why this over Claude code?

[0 replies](#)[Reply](#)

More from Jannis and The Context Layer



Jannis

This Free OpenAI Alternative Is Exploding: Why LocalAI Belongs On Your Desk

Discover how the AI is helping “Create All the Art” in our new series. You’ll think what’s possible for

Welcome back. You are signed into your member account
bg****@jaxondigital.com.

♦ Sep 28



...

In The Context Layer by Jannis

Claude’s New “Skills” Show How Anthropic Is Layering Intelligence on Top of MCP

♦ 3d ago 198 1



...

In The Context Layer by Jannis

Stop Writing Claude Code Configurations From Scratch—This Template

Library C

Welcome back. You are signed into your member account

If you are sp **bg****@jaxondigital.com.**

comprehensive resource that can eliminate this setup...

★ Aug 3 556 3



...

● Jannis

This Free Microsoft AI Curriculum Is the Secret Weapon for 2025—Don't Miss Out!

Unlock real AI skills—step-by-step, hands-on, and totally free.

★ Sep 19 417 9



...

See all from Jannis

See all from The Context Layer

Welcome back. You are signed into your member account
bg****@jaxondigital.com.

Recommended from Medium

 In The Context Layer by Jannis 

Claude’s New “Skills” Show How Anthropic Is Layering Intelligence on Top of MCP

 3d ago  198  1





In nginity by Reza Rezvani

Master Class: How to Turn AI Coding Concepts into Permanent Code

Welcome back. You are signed into your member account
bg****@jaxondigital.com.

Discover how the Anthropic's new tool for Claude Code called "Skills" transform AI Coding assistant from a generic assistant into your...

4d ago 88 2



...



Riccardo Tartaglia

5 Essential MCP Servers Every Developer Should Know

I've been experimenting with Model Context Protocol servers for a few months now, and I have to say, they've changed the way I work.

Oct 11 17 2



...

Welcome back. You are signed into your member account
bg****@jaxondigital.com.

 In AI Software Engineer by Joe Njenga

Claude Haiku 4.5 Released (New) Cheap Model That's Coding Crazy Fast (I Just Tested)

It's been barely days since the release of Claude Sonnet 4.5, and now Anthropic has released Claude Haiku 4.5.

 5d ago  89  1



...

 aakash

Perplexity Just Unleashed 10 FREE AI Agents That Do Your Entire Job (The “Comet” Shortcut)

Stop what you're doing. This is a follow-up AI-generated article from H

Welcome back. You are signed into your member account

[bg****@jaxondigital.com.](#)



...

Oct 7



In Realworld AI Use Cases by Chris Dunlop

Claude Code Tip: You'll waste hours if you don't do this

Do you ever find that you go to use and package and then things just don't work? This article will help with that

6d ago

177

3



...

[See more recommendations](#)