










kleneway / awesome-cursor-mpc-server Public

<> Code Issues 6 Pull requests 4 Actions Projects Security and quality

1 Branch 0 Tags Go to file Go to file <> Code ...

 kleneway Update README.md	90f5809 · last year	
 src	Initial commit with .gitignore	last year
 .gitignore	Initial commit with .gitignore	last year
 LICENSE	Added MIT License	last year
 README.md	Update README.md	last year
 package-lock.json	Initial commit with .gitignore	last year
 package.json	Initial commit with .gitignore	last year
 tsconfig.json	Initial commit with .gitignore	last year

README MIT license

AI Development Assistant MCP Server

Welcome to your AI-powered development toolkit, designed as a Model Context Protocol (MCP) server for Cursor! This project provides intelligent coding assistance through custom AI tools. Note that this is mostly a tutorial demo, and not a production-ready tool.

Features

Code Architect

Call advanced reasoning LLMs to generate plans and instructions for coding agents.

Screenshot Buddy

Take UI design screenshots and use them with the composer agent.

Code Review

Use git diffs to trigger code reviews.


Getting Started

1. Environment Setup

First, you'll need to set up your environment variables. Create a file at `src/env/keys.ts`:

```
export const OPENAI_API_KEY = "your_key_here";  
// Add any other keys you need
```



 **Security Note:** Storing API keys directly in source code is not recommended for production environments. This is only for local development and learning purposes. You can set the env var inline in the Cursor MCP interface as well.

2. Installation

```
npm install  
# or  
yarn install
```



3. Build the Server

```
npm run build
```



4. Adding to Cursor

This project is designed to be used as an MCP server in Cursor. Here's how to set it up:

1. Open Cursor
2. Go to `Cursor Settings > Features > MCP`
3. Click `+ Add New MCP Server`
4. Fill out the form:
 - **Name:** AI Development Assistant
 - **Type:** stdio

- **Command:** `node /path/to/your/project/dist/index.js`

 **Pro Tip:** You might need to use the full path to your project's built index.js file.

After adding the server, you should see your tools listed under "Available Tools". If not, try clicking the refresh button in the top right corner of the MCP server section.

For more details about MCP setup, check out the [Cursor MCP Documentation](#).


Using the Tools

Once configured, you can use these tools directly in Cursor's Composer. The AI will automatically suggest using relevant tools, or you can explicitly request them by name or description.

For example, try typing in Composer:

- "Review this code for best practices"
- "Help me architect a new feature"
- "Analyze this UI screenshot"

The agent will ask for your approval before making any tool calls.

 **Pro Tip:** You can update your `.cursorrules` file with instructions on how to use the tools for certain scenarios, and the agent will use the tools automatically.

Project Structure

```
src/
├─ tools/
│   ├─ architect.ts    # Code structure generator
│   ├─ screenshot.ts  # Screenshot analysis tool
│   └─ codeReview.ts  # Code review tool
├─ env/
│   └─ keys.ts        # Environment configuration (add your API
keys here!)
└─ index.ts          # Main entry point
```

Contributing

Contributions welcome! Please feel free to submit a Pull Request.



License

This project is licensed under the MIT License - see the [LICENSE](#) file for details.



Issues & Support

Found a bug or need help? Open an issue with:

1. What you were trying to do
2. What happened instead
3. Steps to reproduce
4. Your environment details

I'll be honest though, this is a tutorial demo, and not a production-ready tool so I likely won't be fixing issues. But feel free to fork it and make it your own!

Made with by developers, for developers

Releases

No releases published

Packages

No packages published

Contributors 1



kleneway Kevin Leneway

Languages

● TypeScript 100.0%