

54yyyu / code-mcp Public

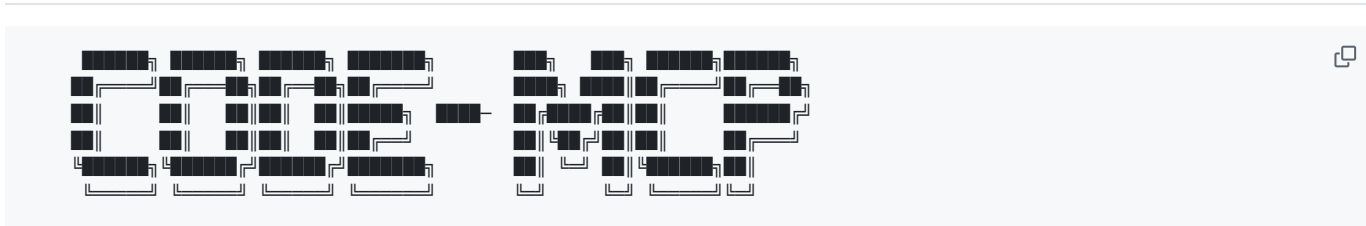
<> Code Issues 3 Pull requests Actions Projects Security and quality Insights

main 4 Branches 0 Tags Go to file Go to file <> Code ...

54yyyu	Add Optional Features section to main README with installation instru...	4cfc464 · last year
src/code_mcp	Revert "Add Remote Connectivity Features"	last year
.gitignore	Improve code replacement with advanced s...	last year
LICENSE	Initial commit: DevTerminal MCP with docu...	last year
README.md	Add Optional Features section to main RE...	last year
install.ps1	minor edits	last year
install.sh	minor edits	last year
main.py	Streamline installation process and clean u...	last year
pyproject.toml	Revert "Add Remote Connectivity Features"	last year
requirements.txt	Fix duplicate dependency in requirements.txt	last year
uninstall.sh	Update installation scripts with correct Clau...	last year

README MIT license

Code-MCP: Terminal and Code Integration for Claude AI



Code-MCP connects Claude AI to your development environment through the Model Context Protocol (MCP), enabling terminal commands and file operations through the AI interface.

Features

- **Terminal Access:** Run shell commands in your project directory directly from Claude
- **File Operations:** Create, read, update, and delete files and directories
- **Git Integration:** Special handling for git commands with safety confirmations
- **Smart Editing:** Enhanced editing capabilities for code files
- **Code Analysis:** Advanced pattern matching and function-level operations
- **Productivity Tools:** Operate on your codebase with natural language instructions
- **Remote Connectivity:** Connect to remote code-mcp instances over SSH

Quick Installation

The following commands install the base version. See the [Optional Features](#) section for installing with additional capabilities.

macOS / Linux

```
# Install with a single command (includes uv installation if needed)
curl -LsSf https://raw.githubusercontent.com/54yyyu/code-mcp/main/install.sh | sh
```

Windows

```
# Download and run the installer
powershell -c "Invoke-WebRequest -Uri https://raw.githubusercontent.com/54yyyu/code-mcp/main/install.ps1 -OutFile ins
```

Manual Installation

```
# Install with pip
pip install git+https://github.com/54yyyu/code-mcp.git

# Or better, install with uv
uv pip install git+https://github.com/54yyyu/code-mcp.git
```

Optional Features

Code-MCP offers optional feature sets that you can choose to install based on your needs:

Remote Connectivity Support

```
# Install with remote connectivity support
pip install git+https://github.com/54yyyu/code-mcp.git@remote-edit

# Or with uv
uv pip install git+https://github.com/54yyyu/code-mcp.git@remote-edit

# One-line installer
curl -LsSf https://raw.githubusercontent.com/54yyyu/code-mcp/remote-edit/install.sh | sh
```

Jupyter Notebook Support

```
# Install with Jupyter notebook support
pip install git+https://github.com/54yyyu/code-mcp.git@jupyter

# Or with uv
uv pip install git+https://github.com/54yyyu/code-mcp.git@jupyter

# One-line installer
curl -LsSf https://raw.githubusercontent.com/54yyyu/code-mcp/jupyter/install.sh | sh
```

Combining Features

To install both Remote Connectivity and Jupyter Notebook support, you can clone the repository and choose which features to include:

```
# Clone the repository
git clone https://github.com/54yyyu/code-mcp.git
cd code-mcp

# Create a custom branch with both features
git checkout main
git merge remote-edit --no-commit
git merge jupyter --no-commit
git commit -m "Custom installation with both remote and jupyter features"
```

```
# Install from your custom branch
pip install -e .
```

For Developers

If you're developing Code-MCP, clone the repository and install in development mode:

```
# Clone the repository
git clone https://github.com/54yyyu/code-mcp.git
cd code-mcp

# Install in development mode with pip
pip install -e .

# Or with uv (recommended)
uv pip install -e .
```

Usage

Claude Desktop Integration

1. Automatic setup:

```
# With a specific project path
code-mcp-setup /path/to/your/project

# Or run without arguments to use the current directory
code-mcp-setup
```

2. Or manually edit Claude Desktop configuration:

- Go to Claude > Settings > Developer > Edit Config
- Add the following to your `claude_desktop_config.json` :

```
{
  "mcpServers": {
    "code": {
      "command": "code-mcp",
      "args": [
        "/path/to/your/project"
      ]
    }
  }
}
```

3. Save the file and restart Claude

Updating Existing Configurations

If you previously installed code-mcp but it requires an absolute path in your configuration, you can use the setup helper with the `--fix-path` flag to update your configuration:

```
code-mcp-setup --fix-path
```

This will find your Claude Desktop configuration and update the command to use just `code-mcp` instead of the full path, allowing it to work as long as code-mcp is in your PATH.

Note: All new installations automatically use the PATH-based approach.

Remote Connectivity

You can connect Claude Desktop to a code-mcp instance running on a remote server.

Prerequisites:

- SSH access to the remote server
- Python 3.10+ on both local and remote machines

The setup script will automatically check if code-mcp is installed on the remote server and install it if needed.

One-line installation

```
# Install and set up remote connection with one command
curl -sSL https://raw.githubusercontent.com/54yyyu/code-mcp/remote-edit/remote-install.sh | bash -s -- --remote-host

# With additional options
curl -sSL https://raw.githubusercontent.com/54yyyu/code-mcp/remote-edit/remote-install.sh | bash -s -- \
  --remote-host user@hostname \
  --remote-project-path /path/to/project \
  --local-port 3000 \
  --remote-port 5000 \
  --ssh-key ~/.ssh/id_ed25519
```

This installs code-mcp locally if needed, configures remote connection, and sets up Claude Desktop.

```
# Set up a remote connection (with default options)
code-mcp-remote --remote-host user@example.com --remote-project-path /path/to/remote/project

# Set up with custom port configuration
code-mcp-remote --remote-host user@example.com \
  --remote-project-path /path/to/remote/project \
  --local-port 3000 \
  --remote-port 5000

# Use a specific SSH key
code-mcp-remote --remote-host user@example.com \
  --remote-project-path /path/to/remote/project \
  --ssh-key ~/.ssh/id_ed25519
```

The remote setup:

1. Checks if code-mcp is installed on the remote server and installs it if needed
2. Uploads the bridge server script to the remote server
3. Starts the bridge server on the remote machine
4. Sets up an SSH tunnel to securely communicate with the remote server
5. Configures Claude Desktop to use the remote connection

This allows you to work with codebases on remote servers, including cloud VMs and containers. When you're done, press Ctrl+C to terminate the SSH tunnel and clean up the configuration.

Features

Terminal Operations

- Execute terminal commands with `run_command()`
- Perform git operations with `git_operation()`

File Operations

- Read files with `read_file()`
- Edit files with `edit_file()` and `smart_edit()`
- Create and delete directories with `create_directory()` and `delete_path()`
- List directory contents with `list_directory()`

Advanced Code Editing

- Flexible pattern matching with whitespace normalization
- Function-level editing with automatic indentation handling
- Batch operations across multiple files with `edit_block()`
- Support for search/replace blocks and unified diff formats

Examples

Ask Claude:

- "List all the files in the current directory"
- "Show me the content of the README.md file"
- "Create a new file called example.py with a simple hello world program"
- "Run git status and show me what files have changed"
- "Make these changes to main.py: replace X with Y"
- "Update the process_data function in data_utils.py to handle null values"

Safety Features

- Confirmation required for destructive operations (delete, overwrite)
- Path safety to prevent operations outside project directory
- Special confirmation for potentially dangerous git commands

Requirements

- Python 3.10 or newer
- Claude Desktop or API access
- Git (optional, for git operations)

Releases

No releases published

Packages

No packages published

Contributors 1



54yyyu Steven Yu

Languages

• Python 86.1% • Shell 10.3% • PowerShell 3.6%