

AI Marketing Platform - Complete Documentation

Table of Contents

1. [Project Overview](#)
2. [MCP Servers Templates Package](#)
3. [MCP Server Quick Reference Card](#)
4. [MCP Server Setup Instructions](#)
5. [Windows PowerShell Commands](#)
6. [macOS/Linux Commands](#)
7. [Common MCP Servers](#)

AI Marketing Platform - Complete Documentation

Table of Contents

- [Project Overview](#)
- [MCP Servers Templates Package](#)
- [MCP Server Quick Reference Card](#)

- [MCP Server Setup Instructions](#)
 - [Windows PowerShell Commands](#)
 - [macOS/Linux Commands](#)
 - [Common MCP Servers](#)
-

Project Overview

aimarketingtool

A SaaS platform for people to use AI to build websites and marketing automation using AI

MCP Servers Templates Package

```
=====
MCP SERVERS TEMPLATES PACKAGE
=====
```

WHAT'S INCLUDED:

- Quick reference cards for daily use
- .mcp.json configuration templates for different use cases
- Environment variable setup guides
- Troubleshooting checklists

- Project setup templates

HOW TO USE THIS PACKAGE:

- Start with folder "1-Quick-Reference" for daily commands
- Use "2-Configuration-Templates" to set up your .mcp.json
- Follow "3-Environment-Setup" for secure API key management
- Check "4-Troubleshooting" if you run into issues
- Use "5-Project-Setup" when starting new projects

QUICK START:

- Copy the .mcp.json template that matches your use case
- Save it in your project root
- Create .env file from .env.example
- Follow the setup guide

NEED HELP?

Post in the Skool community with questions!

Happy coding! 🚀

MCP Server Quick Reference Card

MCP Server Quick Reference Card

For Claude Code Users

Essential Commands

\$3

```
claude mcp add
```

\$3

```
--scope local    → Just you, this project (default)
--scope project  → Team shared via .mcp.json
--scope user     → You, all projects
```

Common Servers

\$3

```
claude mcp add playwright npx @playwright/mcp@latest
```

Use for: Testing, web scraping, screenshots

\$3

```
claude mcp add --transport http stripe https://mcp.stripe.com
```

Use for: Payment processing, customer management

\$3

```
claude mcp add --transport sse linear https://mcp.linear.app/sse
```

Use for: Issue tracking, project planning

\$3

```
claude mcp add --transport http netlify https://netlify-mcp.netlify.app/mcp
```

Use for: Site deployment, DNS management

\$3

```
claude mcp add --transport http notion https://mcp.notion.com/mcp
```

Use for: Knowledge base, documentation

Using MCP Servers in Claude Code

\$3

```
"Show me failed Stripe payments from last week"
```

\$3

```
@stripe:customer://cus_XXXXXX
```

\$3

Type `/` to see available commands

Authentication

Type `/mcp` in Claude Code to authenticate services

Troubleshooting

\$3

- Check configuration exists (.mcp.json or VS Code settings)
- Restart VS Code
- Re-authenticate via `/mcp`
- Check logs in Output panel

\$3

Wrap npx commands with `cmd /c` :

```
claude mcp add playwright cmd /c npx @playwright/mcp@latest
```

\$3

Set environment variable

```
export VAR_NAME=value # macOS/Linux  
$env:VAR_NAME="value" # Windows PowerShell
```

Need help? Post in the Skool community with:

- Your .mcp.json configuration (remove secrets!)
- Error messages
- What you were trying to do

MCP Server Setup Instructions

MCP Server Setup Instructions

Quick Start

- Copy the `.mcp.json` template that matches your use case
- Save it in your project root
- Create `.env` file from `.env.example`
- Set your API keys in `.env`
- Restart VS Code
- Type `/mcp` in Claude Code to authenticate

Files to Copy

FROM templates → TO your project root:

- `example-XXX.mcp.json` → `.mcp.json` (choose one)
- `.env.example` → `.env` (fill in your keys)
- `.gitignore` → `.gitignore` (or append to existing)

What Goes Where

```
your-project/  
├── .mcp.json      ← Copy from templates folder  
├── .env           ← Create from .env.example  
├── .gitignore     ← Include .env in here  
└── [your code files]
```

After Setup

Test each MCP server:

- "Open a browser and navigate to example.com" (Playwright)
- "Show me test customers" (Stripe)
- "Show me open issues" (Linear)

Need Help?

Check `troubleshooting-checklist.pdf` or post in community!

Available Templates

- **example-fullstack.mcp.json** → Web apps with payments & deployment
- **example-agency.mcp.json** → Client management with different credentials
- **example-devops.mcp.json** → Infrastructure & monitoring
- **example-saas.mcp.json** → SaaS product with full stack
- **example-testing.mcp.json** → QA & automated testing

Choose the one that best matches your project type!

Windows PowerShell Commands

```
=====
WINDOWS POWERSHELL COMMANDS FOR MCP SERVERS
=====
```

CHECK IF CLAUDE CLI EXISTS:

```
Get-Command claude
```

ADD MCP SERVER (REMOTE):

```
claude mcp add --transport http stripe https://mcp.stripe.com
```

ADD MCP SERVER (LOCAL - IMPORTANT: Use cmd /c):

```
claude mcp add playwright cmd /c npx @playwright/mcp@latest
```

SET ENVIRONMENT VARIABLE (TEMPORARY):

```
$env:STRIPE_KEY = "sk_test_your_key_here"
```

CHECK ENVIRONMENT VARIABLE:

```
echo $env:STRIPE_KEY
```

LOAD FROM .env FILE:

```
Get-Content .env | ForEach-Object {  
    $name, $value = $_.split('=')  
    Set-Item -Path "env:$name" -Value $value  
}
```

CHECK NODE VERSION:

```
node --version
```

VIEW .mcp.json FILE:

```
type .mcp.json
```

COMMON ISSUE: "Connection closed"

SOLUTION: Always use "cmd /c" before npx on Windows:

```
claude mcp add myserver cmd /c npx @package/name
```

macOS/Linux Commands

```
=====
MACOS/LINUX COMMANDS FOR MCP SERVERS
=====
```

CHECK IF CLAUDE CLI EXISTS:

```
which claude
```

ADD MCP SERVER (REMOTE):

```
claude mcp add --transport http stripe https://mcp.stripe.com
```

ADD MCP SERVER (LOCAL):

```
claude mcp add playwright npx @playwright/mcp@latest
```

SET ENVIRONMENT VARIABLE (TEMPORARY):

```
export STRIPE_KEY="sk_test_your_key_here"
```

SET ENVIRONMENT VARIABLE (PERMANENT):

```
echo 'export STRIPE_KEY="sk_test_your_key_here"' >> ~/.zshrc
source ~/.zshrc
```

CHECK ENVIRONMENT VARIABLE:

```
echo $STRIPE_KEY
```

LOAD FROM .env FILE:

```
source .env
```

CHECK NODE VERSION:

```
node --version
```

VIEW .mcp.json FILE:

```
cat .mcp.json
```

COMMON LOCATIONS:

```
User config: ~/.config/claude-code/mcp_config.json  
Project config: ./mcp.json
```

Common MCP Servers

```
=====
```

COMMON MCP SERVERS - COPY-PASTE READY

```
=====
```

PLAYWRIGHT (Browser Automation):

```
claude mcp add playwright npx @playwright/mcp@latest
```

STRIPE (Payments):

```
claude mcp add --transport http stripe https://mcp.stripe.com
```

LINEAR (Project Management):

```
claude mcp add --transport sse linear https://mcp.linear.app/sse
```

NETLIFY (Deployment):

```
claude mcp add --transport http netlify https://netlify-mcp.netlify.app/mcp
```

NOTION (Documentation):

```
claude mcp add --transport http notion https://mcp.notion.com/mcp
```

SENTRY (Error Tracking):

```
claude mcp add --transport http sentry https://mcp.sentry.dev/mcp
```

VERCEL (Deployment):

```
claude mcp add --transport http vercel https://mcp.vercel.com/
```

ASANA (Project Management):

```
claude mcp add --transport sse asana https://mcp.asana.com/sse
```

HUBSPOT (CRM):

```
claude mcp add --transport http hubspot https://mcp.hubspot.com/anthropic
```

INTERCOM (Customer Support):

```
claude mcp add --transport http intercom https://mcp.intercom.com/mcp
```

=====

WITH PROJECT SCOPE (Team Sharing):

=====

Add --scope project before server name:

```
claude mcp add --scope project playwright npx @playwright/mcp@latest
```

This creates .mcp.json file you can commit to Git.

End of Documentation

Generated on: \$(date)

This document combines all project documentation into a single comprehensive guide.