

# AI Marketing Platform – MCP Servers and Setup

---

## Table of Contents

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

## 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:

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

### QUICK START:

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

NEED HELP?

Post in the Skool community with questions!

Happy coding! 🚀

---

# MCP Server Quick Reference Card

*For Claude Code Users*

## Essential Commands

### Add MCP Server

```
claude mcp add <name> <command> <args>
```

### Scopes

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

## Common Servers

### Playwright (Browser Automation)

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

**Use for:** Testing, web scraping, screenshots

### Stripe (Payments)

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

**Use for:** Payment processing, customer management

## Linear (Project Management)

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

**Use for:** Issue tracking, project planning

## Netlify (Deployment)

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

**Use for:** Site deployment, DNS management

## Notion (Documentation)

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

**Use for:** Knowledge base, documentation

# Using MCP Servers in Claude Code

## Method 1: Natural Language

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

## Method 2: @ Mentions

```
@stripe:customer://cus_XXXXX
```

## Method 3: / Commands

Type `/` to see available commands

# Authentication

Type `/mcp` in Claude Code to authenticate services

## Troubleshooting

### Server Not Working?

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

### Windows Users

Wrap npx commands with `cmd /c` :

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

### Environment Variables Not Loading?

```
# 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

### Quick Start

1. Copy the .mcp.json template that matches your use case
2. Save it in your project root
3. Create .env file from .env.example
4. Set your API keys in .env

5. Restart VS Code
6. 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

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

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

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.*