MCP Servers Configured

Current Configuration

File: `.cursor/mcp.json`

Active MCP Servers

1. Filesystem MCP ■

Purpose: File system operations
Status: Active
Useful for:
- Reading/writing files
- Listing directories
- Searching files
- File metadata

2. GitHub MCP ■

Purpose: Access GitHub repositories
Status: Configured (needs API key)
Useful for:
- Reading moOde source code
- Checking for updates
- Accessing issues/PRs

Setup: Add `GITHUB`*`PERSONAL`*`ACCESS_TOKEN` to `.env`

3. Brave Search MCP ■

Purpose: Web search
Status: Configured (needs API key)
Useful for:
- Searching moOde forum
- Finding solutions
- Research

Setup: Add `BRAVE`*`API`*`KEY` to `.env`

4. PostgreSQL MCP ■

Purpose: PostgreSQL database access
Status: Configured (needs connection string)
Useful for:
- Database operations (if moOde uses PostgreSQL)

Setup: Add `POSTGRES`*`CONNECTION`*`STRING` to `.env`

5. SQLite MCP ■

Purpose: SQLite database access
Status: Configured

Database path: `moode-source/var/local/www/db/moode-sqlite3.db`
Useful for:
• Reading moOde configuration
• Querying system settings
• Updating configuration
• Checking audio settings

6. Puppeteer MCP ■

Purpose: Browser automation
Status: Active
Useful for:
• Testing moOde Web UI
• Taking screenshots
• Automating browser tasks
• Testing Room Correction Wizard

7. Fetch MCP ■

Purpose: HTTP requests
Status: Active
Useful for:
• API calls
• Testing endpoints
• HTTP requests to Pi

8. Memory MCP ■

Purpose: Persistent memory
Status: Active
Useful for:
• Remembering context
• Storing information
• Learning from interactions

Setup Instructions

For GitHub MCP

1. Create GitHub Personal Access Token
2. Add to `.env`: `GITHUB_PERSONAL_ACCESS_TOKEN=your_token`

For Brave Search MCP

1. Get Brave Search API key
2. Add to `.env`: `BRAVE_API_KEY=your_key`

For PostgreSQL MCP

1. If needed, add connection string to `.env`
2. Format: `POSTGRES_CONNECTION_STRING=postgresql://user:pass@host/db`

Usage

SQLite MCP - moOde Database

```
# I can now:
- Query moOde database
- Read configuration
- Update settings
- Check system state
```

Puppeteer MCP - Browser Automation

```
# I can now:
- Navigate to moOde Web UI
- Take screenshots
- Click buttons
- Test functionality
- Automate Room Correction Wizard
```

Fetch MCP - HTTP Requests

```
# I can now:
- Test moOde API endpoints
- Check Web UI status
- Make HTTP requests
```

Benefits

1. **Automated Testing** - Browser MCP can test Web UI
2. **Database Access** - SQLite MCP can read moOde config
3. **Web Search** - Brave Search can find solutions
4. **GitHub Access** - Read source code directly
5. **HTTP Testing** - Fetch MCP can test APIs

Next Steps

1. **Use SQLite MCP** to read moOde database
2. **Use Puppeteer MCP** to test Web UI
3. **Use Fetch MCP** to test API endpoints
4. **Add API keys** if needed for GitHub/Brave Search