QRCodeKIT

INSTALLATION GUIDE

Connect QRCodeKIT MCP to your *AI agent*.

Follow the steps for the AI client you use to add the QRCodeKIT MCP server and start creating and managing QR codes through plain language prompts.

VERSION	UPDATED	SUPPORTED CLIENTS
1.0	**May 2026**	**Claude, ChatGPT, Cursor, VS Code, Windsurf**

Before you start

QRCodeKIT MCP is a server that connects your QRCodeKIT account to any AI agent that supports the Model Context Protocol. Once configured, you can create, list, view, update and delete dynamic QR codes through plain language prompts.

MCP SERVER URL

```
https://mcp.v2.qrcodekit.com/mcp
```

You will need:

1. A QRCodeKIT account. If you don't have one yet, sign up at `qrcodekit.com`.
2. An MCP-compatible AI client (see the table below).
3. A few minutes to paste the server URL and authorize the connection.

Recommended setup by client

CLIENT	RECOMMENDED SETUP	NOTES
ChatGPT	OAuth direct HTTP connection	Best fit for the hosted MCP server and QR gallery widget.
Claude Web	OAuth direct HTTP connection	Supported by the production deployment's allowed browser origins.
Claude Code	HTTP connection with OAuth if supported	For manual bearer-token testing, use the mcp-remote example below.
Cursor	HTTP connection with OAuth if supported	Use the hosted MCP URL.
VS Code	HTTP connection with OAuth if supported	Use the hosted MCP URL.
Windsurf	mcp-remote bridge when needed	

CLIENT	RECOMMENDED SETUP	NOTES
		Useful for clients that need a local stdio bridge to a remote MCP server.
Other HTTP-compatible clients	OAuth direct HTTP connection	The client must support streamable HTTP and OAuth protected-resource discovery.

ChatGPT `OAUTH`

1. In ChatGPT, go to **Settings > Apps & Connectors > Advanced Settings**.
2. Turn on **Developer Mode**.
3. Go back to **Apps & Connectors** and choose **Create**.
4. Add the QRCodeKIT MCP Server with:
   Server URL: `https://mcp.v2.qrcodekit.com/mcp`
   Authentication: `OAuth`
5. When prompted, sign in to QRCodeKIT and authorize the requested QR scopes.

After authorization, ChatGPT can call QRCodeKIT tools for the connected account.

Claude Web `OAUTH`

1. Open your Claude workspace settings.
2. Go to the MCP server configuration area.
3. Add the QRCodeKIT MCP Server with:
   Server URL: `https://mcp.v2.qrcodekit.com/mcp`
   Authentication: `OAuth`
4. When prompted, sign in to QRCodeKIT and authorize the connection.

Claude Code HTTP · OAUTH

If your Claude Code version supports remote HTTP MCP servers with OAuth, add the server to `~/.claude/mcp.json`:

```
{
  "mcpServers": {
    "QRCodeKit": {
      "type": "http",
      "url": "https://mcp.v2.qrcodekit.com/mcp"
    }
  }
}
```

Restart Claude Code after saving the file.

> **Advanced testing.** For local testing with an existing OAuth access token, use `mcp-remote`:

```
{
  "mcpServers": {
    "QRCodeKit": {
      "command": "npx",
      "args": [
        "mcp-remote",
        "https://mcp.v2.qrcodekit.com/mcp",
        "--header",
        "Authorization: Bearer ${QRCODEKIT_ACCESS_TOKEN}"
      ],
      "env": {
        "QRCODEKIT_ACCESS_TOKEN": "YOUR_OAUTH_ACCESS_TOKEN"
      }
    }
  }
}
```

Prerequisite: Node.js 20 LTS or higher with npx support.

Cursor HTTP · OAUTH

If your Cursor version supports remote HTTP MCP servers with OAuth, add the server to `~/.cursor/mcp.json`:

```
{
  "mcpServers": {
    "QRCodeKit": {
      "type": "http",
      "url": "https://mcp.v2.qrcodekit.com/mcp"
    }
  }
}
```

Restart Cursor after saving the file.

VS Code HTTP · OAUTH

Add the server to `.vscode/mcp.json` in your workspace:

```
{
  "servers": {
    "QRCodeKit": {
      "type": "http",
      "url": "https://mcp.v2.qrcodekit.com/mcp"
    }
  }
}
```

Restart VS Code or reload the window after saving the file.

Windsurf MCP-REMOTE BRIDGE

For clients that require a local stdio bridge, use `mcp-remote` with an OAuth access token:

```
{
  "mcpServers": {
    "QRCodeKit": {
      "command": "npx",
      "args": [
        "mcp-remote",
        "https://mcp.v2.qrcodekit.com/mcp",
        "--header",
        "Authorization: Bearer ${QRCODEKIT_ACCESS_TOKEN}"
      ],
      "env": {
        "QRCODEKIT_ACCESS_TOKEN": "YOUR_OAUTH_ACCESS_TOKEN"
      }
    }
  }
}
```

Prefer direct OAuth configuration when the client supports it.

Test it out

Once configured, try asking your AI client:

```
› What QRCodeKIT tools are available?
```

```
› Show my QR codes.
```

```
› Create a QR called "Example campaign" that points to https://
example.com.
```

To update an existing QR, first ask the client to list your QR codes, then use the returned QR ID:

```
› Update the QR with id <qr_id> so it points to https://example.com/
spring-campaign.
```

You're all set. *Start prompting.*

QRCodeKIT MCP is now connected to your AI agent. Ask anything about your QR codes and let the agent handle the rest.

Need help? Reach out at support@qrcodekit.com
Learn more at qrcodekit.com/ai-qr-codes/mcp
