n8n MCP Server with Gemini CLI

Quick Guide (CLI)

1. Installation of the n8n MCP Server

To take full advantage of the n8n MCP Server you first need to have:

- a stable internet connection
- node.js already installed
- Gemini CLI already installed and authenticated
- Docker **<- Optional**

If you don't have node.js and the Gemini CLI installed, refer back to my previous guide about Installing Gemini CLI on your device.

⚠️ ***Be mindful: There is no official documentation talking about the support of n8n with the Gemini CLI. If something doesn't work you only have this and my other publications!***

OPTIONAL: How to install Docker
If you chose to install the n8n MCP Server using Docker you will first need to download the Docker Desktop software before starting the n8n MCP installation.
Here is how to install Docker using Terminal commands:
—----------

ON WINDOWS:
> winget install -e –id Docker.DockerDesktop
—----------

ON LINUX:
> curl -fsSL https://get.docker.com | sudo sh
> sudo usermod -aG docker $USER

ON MAC:
> brew install --cask docker
—----------

When you have it installed check if it installed correctly by restarting the terminal and using the following command in the terminal:
> docker -v

If your terminal shows this after using that command:

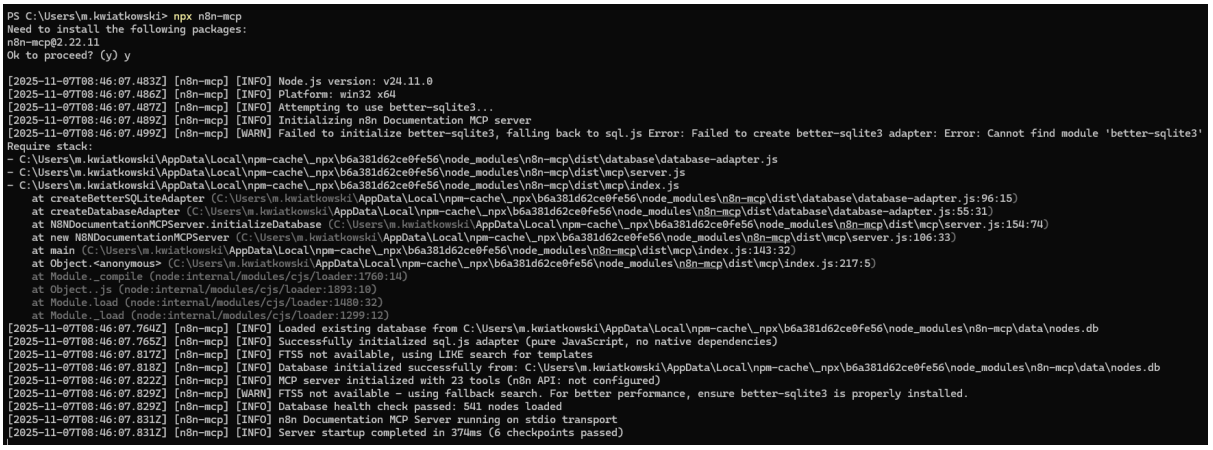
```
PS C:\Users\toddg> docker -v
Docker version 28.5.1, build e180ab8
```

You have successfully installed Docker on your device, but if something went wrong restart your terminal and try again or refer to the Official Documentation Of Docker.

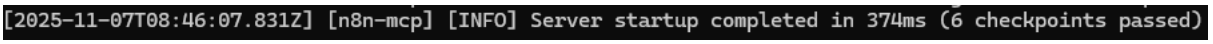
When wanting to use the n8n MCP Server we have 2 options to work with:

1. npx - (Fastest - No installation)

This option doesn't even include the installation of the n8n MCP Server because it runs directly through npx. To run it you need to use the following command:
> npx n8n-mcp
If everything goes according to plan you should see something along the lines of this on your screen:



If there are no errors the previous screen should end in this line that indicates the server is now up and running.



To make the MCP Server work with the Gemini CLI you need to import the MCP Server into Geminis settings.
You can find the file under this path:
ON WINDOWS:
> C:\Users\<your_username>\.gemini\settings.json
ON LINUX/MAC:
> ~/.gemini/settings.json

To import the n8n MCP Server correctly need to use the following configuration code and paste it in the settings.json file.

```
"mcpServers": {
    "n8n-mcp": {
      "command": "npx",
      "args": ["n8n-mcp"],
      "env": {
        "MCP_MODE": "stdio",
        "LOG_LEVEL": "error",
        "DISABLE_CONSOLE_OUTPUT": "true",
        "N8N_API_URL": "https://your-n8n-instance.com",
        "N8N_API_KEY": "your-api-key"
      }
    }
}
```

REMEMBER TO FILL IN YOUR API KEY AND YOUR API URL WITH CORRECT DATA!

After all of that you can now run the Gemini CLI with the “gemini” command in the terminal.
To check if the n8n MCP Server got imported correctly, in the Gemini CLI use the following command: *> /mcp reload*

If everything went according to plan you should see this on your screen:

Congratulations now you can take advantage of the n8n MCP Server with the Gemini CLI!

2. Docker (Easy & Isolated)

This option includes installing the n8n MCP Server using Docker.
Firstly you need to pull the docker image using the following command: > *docker pull* ghcr.io/czlonkowski/n8n-mcp:latest
Then to run it just turn on your container.
To make the MCP Server work with the Gemini CLI you need to import the MCP Server into Geminis settings.
You can find the file under this path:
ON WINDOWS:
> *C:\Users\<your_username>\.gemini\settings.json*
ON LINUX/MAC:
> *~/.gemini/settings.json*
To import the n8n MCP Server correctly need to use the following configuration code and paste it in the settings.json file.

```
{
  "mcpServers": {
    "n8n-mcp": {
      "command": "docker",
      "args": [
        "run",
        "-i",
        "--rm",
        "--init",
        "-e", "MCP_MODE=stdio",
        "-e", "LOG_LEVEL=error",
        "-e", "DISABLE_CONSOLE_OUTPUT=true",
        "-e", "N8N_API_URL=https://your-n8n-instance.com",
        "-e", "N8N_API_KEY=your-api-key",
        "ghcr.io/czlonkowski/n8n-mcp:latest"
      ]
    }
  }
}
```

REMEMBER TO FILL IN YOUR API KEY AND YOUR API URL WITH CORRECT DATA!

After all of that you can now run the Gemini CLI with the “gemini” command in the terminal.
To check if the n8n MCP Server got imported correctly, in the Gemini CLI use the following command:
> /mcp reload
If everything went according to plan you should see this on your screen:

NOW YOUR GEMINI CLI CAN WORK WITH THE n8n tools. HAVE FUN AND BE CAREFUL!