Bind Mount, Dev Container & Ts Node Dev Cheat Sheet

Bind Mound Syntax For

💡 For Git Bash

```
docker run -p 5000:5000 --name ts-container -w //app
-v ts-docker-logs://app/logs -v "//$(pwd)"://app/ -v
//app/node_modules --rm ts-docker
```

💡 For Powershell

```
docker run -p 5000:5000 --name ts-container -w //app
-v ts-docker-logs://app/logs -v "${PWD}://app" -v //a
pp/node_modules --rm ts-docker
```

💡 For CMD

```
docker run -p 5000:5000 --name ts-container -w //app
-v ts-docker-logs://app/logs -v "%cd%"://app/ -v //ap
p/node_modules --rm ts-docker
```

ts-node-dev command for Docker Container

> 💡 ts-node-dev --respawn --transpile-only --poll src/server.ts

1. First Open A .devcontainer folder in the root of the project
2. Inside the .devcontainer folder open a file named devcontainer.json and paste the following code
3. Change the json name, container names and file directories according to your project

```
{
  "name": "ts-container",
  "image": "node:20",
  "workspaceFolder": "/app",
  "mounts": [
    // Bind mount for your local project
    "source=/c/Projects/next-level/Docker/docker-with-typescipt-backend,target=/app,type=bind",

    // Named volume for logs (similar to: -v ts-docker-logs://app/logs)
    "source=ts-docker-logs,target=/app/logs,type=volume",

    // Anonymous volume for node_modules (similar to: -v //app/node_modules)
    "target=/app/node_modules,type=volume"
  ],
  "runArgs": [
    "--name",
    "ts-container",
    "-p",
    "5000:5000",
    "--rm" // Automatically remove the container after exiting VS Code
  ],
  "postCreateCommand": "npm install"
}
```

Docker container run (with container name) and named volume

```
docker run -p 5000:5000 --name docker-volume-container --rm -v docker-logs:/app/logs docker-volume:v2
```

- -v docker-logs:/app/logs = -v for volume; with volume name before the semicolon

Binding Mount Run

Step 1 to run (Powershell)

```
docker run -p 5000:5000 --name docker-volume-container -v docker-logs:/app/logs -w /app -v "E:/Level 2 Reward/Cloud-Container Craftsman/Module 3/docker-with-typescript-backend:/app" -v /app/node_modules --rm docker-volume:v2
```

- the first -v = named volume; naming a volume
- this is binding method so in command we have to give the working directory -w /app
- the second -v = (for binding method) the absolute path in the “”
- the third -v = (anonymous volume) → here the node_modules code doesn’t replace the volume code
- In docker the more depth the path → the more priority it has

Step 2 to run (Powershell)

```
docker run -p 5000:5000 --name docker-volume-container -v docker-logs:/app/logs -w /app -v "$(pwd)" -v /app/node_modules -
```

```
-rm docker-volume:v2
```

- $(pwd) = print working directory

We use this instead of the absolute path