Status  
VS Terminal works:

.\debug-testMaster.cmd

VS Terminal doesn’t work

.\debug-testMaster-Universal.cmd utils/testFetchData.ts

.\debug-testMaster-Universal.cmd utils/testMaster.ts

Perfect — this is a **very well-structured launch.json**, and the fact that

.\debug-testMaster-Universal.cmd utils/testMaster.ts

✅ runs fine in the terminal tells us the scripts themselves are solid.

The “**runs endlessly in VS Code debugger**” symptom is actually caused by the **debug type** you’re using — node-terminal.

Let’s fix that. 👇

**⚙️ Why it hangs**

"type": "node-terminal" in VS Code does **not** attach the debugger to the process — it just spawns a terminal.  
When the .cmd runs and finishes, VS Code doesn’t detect that the process ended (especially when node --inspect or other loaders are used).  
So the Debug session never reports “done”.

You basically launched a terminal *inside* the debugger, not a Node process under debug control.

**✅ 2 Fix Options**

**Option A — change to type: "node"**

Let’s make VS Code **actually attach** the Node debugger directly to your runner.

Replace your configuration with this one:

{

"name": "5b.Debug Specific TS Test - parameterized",

"type": "node",

"request": "launch",

"program": "${workspaceFolder}/run-test.mjs",

"args": ["${input:testFile}"],

"runtimeArgs": [

"--no-warnings",

"--loader",

"ts-node/esm"

],

"cwd": "${workspaceFolder}",

"env": {

"TS\_NODE\_PROJECT": "tsconfig.node.json"

},

"console": "integratedTerminal",

"skipFiles": ["<node\_internals>/\*\*"],

"sourceMaps": true

}

✅ Explanation:

* It runs **Node** directly (type: "node").
* Launches your universal test runner (run-test.mjs).
* Passes your chosen test file as argument.
* The debugger attaches correctly and exits when your test ends.

When you hit **F5**, you’ll get the prompt:

Enter path to test file (e.g. utils/testLogger.ts)

Type (or press Enter for default):

utils/testMaster.ts

Then you’ll see:

🧪 Running test file: file:///D:/Dev/.../utils/testMaster.ts

and the debugger will stop once the script finishes.

**Option B — stay with node-terminal but auto-close**

If you prefer to keep type: "node-terminal" (because it runs in a normal terminal),  
you can make the .cmd exit cleanly at the end with:

:: debug-testMaster-Universal.cmd

@echo off

setlocal

set TEST\_FILE=%1

if "%TEST\_FILE%"=="" set TEST\_FILE=utils/testMaster.ts

echo Running test file: %TEST\_FILE%

node --no-warnings --loader ts-node/esm run-test.mjs %TEST\_FILE%

echo.

echo ✅ Done. Press any key to close.

pause >nul

exit

Then VS Code will close the session when you press a key.

**💡 TL;DR — Recommendation**

✅ Use **Option A (type: "node")** — it’s the *true* debugger experience.  
You can set breakpoints in your .ts files, step into code, inspect variables, and it will end automatically when the test finishes.

Would you like me to extend this so you can pick your test from a **dropdown menu** instead of typing it in each time?

Top of Form

Bottom of Form

ChatGPT kann Fehler machen. Überprüfe wichtige Informationen. Si