**Node project opzetten met Jest in debug**

1. Maak een JS file aan met wat ES6 syntax
2. Run *npm init -y*
3. Voeg in de aangemaakte package.json “type”:”module” toe
4. Run *npm install node*
5. Run *npm install --save-dev jest*
6. Maak een Jest JS file aan met een ES6 import statement
7. Run  *node --experimental-vm-modules node\_modules/jest/bin/jest.js* om te controleren of Jest runt met ES6 syntax. Jest runt maar nu willen we Jest nog kunnen debuggen
8. Ga naar <https://github.com/microsoft/vscode-recipes/tree/main/debugging-jest-tests> en kopieer de volgende launch.json configuratie:

{

"type": "node",

"request": "launch",

"name": "Jest Current File",

"program": "${workspaceFolder}/node\_modules/.bin/jest",

"args": [

"--runTestsByPath",

"${relativeFile}",

"--config",

"jest.config.js"

],

"console": "integratedTerminal",

"internalConsoleOptions": "neverOpen",

"disableOptimisticBPs": true,

"windows": {

"program": "${workspaceFolder}/node\_modules/jest/bin/jest",

}

}

1. Klik nu op *Run & debug*  in VS Code en vervolgens op *create a launch.json file* (selecteer hierbij Node.Js)
2. Vervang de configuratie in de launch.json file door de configuratie die we hebben geselecteerd. Let op: vervang alleen het gedeelte binnen “configuration”
3. Verander in de launch.json *jest.config.js* naar *jest.config.ts*
4. Maak in de working directory een *jest.config.ts* file aan met de volgende inhoud:  
     
   import type { Config } from '@jest/types'

const config: Config.InitialOptions = {

    preset: 'ts-jest',

    testEnvironment: 'node',

    verbose: true

}

export default config;

1. Zet een aantal breakpoint in je Jest test file en je reguliere JS file
2. Run npm install ts-node --save-dev
3. npm install ts-jest --save-dev
4. Open het tabblad met je Jest test file, klik op *Run and Debug* en klik op de groene play button met *Jest current file*

<https://www.youtube.com/watch?v=_usf3xQ7wys>