

Debugging Tests

There are various ways to setup a debugger for your Jest tests. We cover debugging in Chrome and Visual Studio Code.

Debugging Tests in Chrome

Add the following to the `scripts` section in your project's `package.json`

```
"scripts": {  
  "test:debug": "react-scripts --inspect-brk test --runInBand --no-cache"  
}
```

Place `debugger;` statements in any test and run:

```
$ npm run test:debug
```

This will start running your Jest tests, but pause before executing to allow a debugger to attach to the process.

Open the following in Chrome

```
about:inspect
```

After opening that link, the Chrome Developer Tools will be displayed. Select `inspect` on your process and a breakpoint will be set at the first line of the react script (this is done to give you time to open the developer tools and to prevent Jest from executing before you have time to do so). Click the button that looks like a “play” button in the upper right hand side of the screen to continue execution. When Jest executes the test that contains the debugger statement, execution will pause and you can examine the current scope and call stack.

Note: the `--runInBand` cli option makes sure Jest runs test in the same process rather than spawning processes for individual tests. Normally Jest parallelizes test runs across processes but it is hard to debug many processes at the same time.

Debugging Tests in Visual Studio Code

Debugging Jest tests is supported out of the box for Visual Studio Code.

Use the following `launch.json` configuration file:

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "Debug CRA Tests",
      "type": "node",
      "request": "launch",
      "runtimeExecutable": "${workspaceRoot}/node_modules/.bin/react-scripts",
      "args": ["test", "--runInBand", "--no-cache", "--watchAll=false"],
      "cwd": "${workspaceRoot}",
      "protocol": "inspector",
      "console": "integratedTerminal",
      "internalConsoleOptions": "neverOpen",
      "env": { "CI": "true" },
      "disableOptimisticBPs": true
    }
  ]
}
```