

# LeakPair

Proactive repairing of memory leaks in Single Page Applications

## Installation and Usage

1. Since *LeakPair* is an npx command tool, the only requirement for running it is to ensure you have Node installed in your system. *LeakPair* support node versions `>= 14`.

- **On Windows and Mac**

Visit the official [Node](#) website and follow the instructions for your OS. This will install the latest version of Node on your system. This will also install the Node Package Manager (npm) and the node packages executor (npx) on your system. Check their successful installation with `node -v`, `npm -v`, and `npx -v` respectively.

- **On Linux based systems**

Use the command `sudo apt install npm`. This will automatically install node, npm, and npx. However, the installed node version will be `v12.22.9`, while *LeakPair* require at least version 14. For that, install a node version manager globally on your system: `npm install -g n`.

Now you can use `n 14.0.0` to install the supported version. Note that you can replace this version with any version higher than 14, including `n latest`.

2. Once the supported version of node is installed, you can run *LeakPair* with the command `npx leakpair [path]` where `path` is the **absolute** path of the repository or file that you wish to repair for memory leaks. Note that *LeakPair* only supports `.js`, `.jsx`, `.ts` and `.tsx` files. That is, if you provide the path to a repository, it will only filter and process files with these extensions.
3. By default, *LeakPair* will log the memory leak and repair statistics directly in the CLI. However, if you wish to store it in a file format, you can provide an optional argument for the output path: `npx leakpair [path-of-project-to-repair] [path-for-output]`

**Example** `npx leakpair /Users/arooba/roosterjs /Users/arooba/Desktop`

In this case, a `leakpair_results.json` file will be generated containing the leaks and repair statistics of roosterjs, in the user's Desktop directory.

**Example of *LeakPair*'s output log in CLI**

All done.  
Results:  
0 errors  
453 unmodified  
0 skipped  
17 ok  
Time elapsed: 7.652seconds

### Memory leak fix results

#### Refactored Files

##### Refactored for potential timing event leaks

```
Set(6) {  
  'cncjs/src/app/components/RepeatButton/index.jsx',  
  'cncjs/src/app/widgets/Visualizer/WatchDirectory.jsx',  
  'cncjs/src/app/widgets/Visualizer/index.jsx',  
  'cncjs/src/app/widgets/Webcam/Webcam.jsx',  
  'cncjs/src/app/containers/Workspace/Workspace.jsx',  
  'cncjs/src/app/widgets/Connection/index.jsx'  
}
```

##### Refactored for potential event listener leaks

```
Set(2) {  
  'cncjs/src/app/widgets/Visualizer/Visualizer.jsx',  
  'cncjs/src/app/widgets/Custom/Custom.jsx'  
}
```

##### Refactored for potential animation frame request leaks

```
Set(1) { 'cncjs/src/app/widgets/Visualizer/Visualizer.jsx' }
```

Total files refactored 8

Total uncleared timing events detected 8

Total unremoved event listeners detected 4

Total uncanceled animation frame requests detected 3

Total count of potential leaks 15