

Wesley Grimes



# Automatically Remove All Unused Imports in a TypeScript Project

Feb 14, 2019 • 2 min read • Wesley Grimes



## The Problem

Recently, I had a need arise to programatically and recursively traverse through all `*.ts` files in a given project and remove all unused TypeScript imports. At the time this article was written, there was no way to do this within Visual Studio Code without opening each individual `*.ts` file and hitting `CTRL + Shift + O` on Windows/Linux. After some research, and much appreciated help from Twitter colleagues, I found a solution that works. This should work for Angular, React, Vue.js, or any plain TypeScript project.

# The Solution

We will use the `tslint` command line tool, in conjunction with the `tslint-etc` rules, to automatically detect and remove all unused imports in the directory, recursively. If you have a large project, the process can take some time to run. It is important to double-check all files for correctness once the fix process is complete.

## Install Required Tools

Install the follow node packages required to for this process to work.

```
$ npm install -g typescript tslint tslint-etc
```

## Create a temporary `tslint` config file

Create a new file named `tslint-imports.json` in the root of your project. This creates a hyper-focused tslint process that will only check for unused declarations. It is *important to note* that this will throw `tslint` errors on unused imports, parameters and variables. We are only using this for the `--fix` process. As such, the `tslint-etc` rules only autofix unused imports.

This file needs the following contents:

```
{
  "extends": [
    "tslint-etc"
  ],
  "rules": {
    "no-unused-declaration": true
  }
}
```

## Run the autofix process

The following command will traverse, recursively through all `*.ts` files in the project and remove the unused imports. It save the files automatically in place.

**BE CAREFUL!** This process is only reversible if you are using a source control solution like `git` or `svn`, where you can revert changes.


```
$ tslint --config tslint-imports.json --fix --project .
```

## Double-Check Your Files

At this point, I would highly recommend you double-check your files for correctness. This tool worked on all but 2 of my 195 `*.ts` files. Two of the components were incorrectly updated. I was able to spot this by running an `ng build --prod` as it was an `Angular` application. You could run a manual `tslint --project .` if your project doesn't use a build tool.

## Resources

- [Original Twitter Conversation](#)
- [TSLint](#)
- [tslint-etc](#)

**0 Comments**   **Wesley Grimes - Blog****Login**  **Recommend** 5    **Tweet**    **Share****Sort by Best** 

Start the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS 

Be the first to comment.

## ALSO ON WESLEY GRIMES - BLOG

**Angular Routing - Best Practices for Enterprise Applications**

1 comment • 5 months ago

**Oyedepo Olufemi** — Great article Wes! Just what ive been looking for...**NgRx—Best Practices for Enterprise Angular Applications**


6 comments • 5 months ago

**duard** — Hi Wes Grimes , you could re-write this article with the new ngrx v8 :-)**Building an AOT Friendly Dynamic Content Outlet in Angular**

2 comments • 5 months ago

**Wesley Grimes** — Not yet. With ivy enabled in next version of angular you should be able to.**Making Upgrades to Angular's NgOnDestroy | Wesley Grimes**

1 comment • 4 months ago

**Yevheniy Potupa** — I also had issues with unsubscribe method. But, while using takeUntil and takeWhile Observable methods with ... **Subscribe**    **Add Disqus to your site** **Add Disqus** **Add**    **Disqus' Privacy Policy** **Privacy Policy** **Privacy Policy**

## Contact Info

Wesley Grimes

[wesgrimes@gmail.com](mailto:wesgrimes@gmail.com)

Wes is a lead software engineer specializing in enterprise web apps. In his spare time, he loves mentoring new developers, tracking the weather, and contributing to open source projects like NgRx. If you follow him on Twitter you're going to find a lot of pictures of food, sunsets, and tidbits about Angular.

