

```
PS C:\Users\Chee Hong\Desktop\CS3281 - Thematic Systems Project I\CATcher> npm i
```

```
npm ERR! code ERESOLVE
```

```
npm ERR! ERESOLVE could not resolve
```

```
npm ERR!
```

```
npm ERR! While resolving: apollo-angular@2.6.0
```

```
npm ERR! Found: @angular/core@13.4.0
```

```
npm ERR! node_modules/@angular/core
```

```
npm ERR!   @angular/core@"^13.4.0" from the root project
```

```
npm ERR!   peer @angular/core@"13.4.0" from @angular/animations@13.4.0
```

```
npm ERR!   node_modules/@angular/animations
```

```
npm ERR!     @angular/animations@"^13.4.0" from the root project
```

```
npm ERR!     peer @angular/animations@"^13.0.0 || ^14.0.0-0" from @angular/material@13.3.9
```

```
npm ERR!     node_modules/@angular/material
```

```
npm ERR!       @angular/material@"^13.3.9" from the root project
```

```
npm ERR!       1 more (ngx-mat-select-search)
```

```
npm ERR!       1 more @angular/platform-browser
```

```
npm ERR!       8 more (@angular/cdk, @angular/common, @angular/forms, ...)
```

```
npm ERR!
```

```
npm ERR! Could not resolve dependency.
```

```
npm ERR! peer @angular/core@"^6.1.0 || ^7.0.0 || ^8.0.0 || ^9.0.0 || ^10.0.0 || ^11.0.0 || ^12.0.0" from apollo-an
```

```
npm ERR! node_modules/apollo-angular
```

```
npm ERR!   apollo-angular@"^2.6.0" from the root project
```

```
npm ERR!
```

```
npm ERR! Conflicting peer dependency: @angular/core@12.2.17
```

```
npm ERR! node_modules/@angular/core
```

```
npm ERR!   peer @angular/core@"^6.1.0 || ^7.0.0 || ^8.0.0 || ^9.0.0 || ^10.0.0 || ^11.0.0 || ^12.0.0" from apollo-
```

```
npm ERR!   node_modules/apollo-angular
```

```
npm ERR!     apollo-angular@"^2.6.0" from the root project
```

```
npm ERR!
```

Escaping Dependency Hell

(an ongoing story)

By Wong Chee Hong

PS C:\Users\Chee Hong\Desktop\CS3281 - Thematic Systems Project I\CATcher> npm i

⊗ npm ERR! code ERESOLVE

npm ERR! ERESOLVE could not resolve

npm ERR!

npm ERR! While resolving: **apollo-angular@2.6.0**

npm ERR! Found: **@angular/core@13.4.0**

npm ERR! node_modules/@angular/core

npm ERR! **@angular/core@"^13.4.0"** from the root project

npm ERR! peer **@angular/core@"13.4.0"** from **@angular/animations@13.4.0**

npm ERR! node_modules/@angular/animations

npm ERR! **@angular/animations@"^13.4.0"** from the root project

npm ERR! peer **@angular/animations@"^13.0.0 || ^14.0.0-0"** from **@angular/material@13.3.9**

npm ERR! node_modules/@angular/material

npm ERR! **@angular/material@"^13.3.9"** from the root project

npm ERR! 1 more (ngx-mat-select-search)

npm ERR! 1 more (@angular/platform-browser)

npm ERR! 8 more (@angular/cdk, @angular/common, @angular/forms, ...)

npm ERR!

npm ERR! Could not resolve dependency:

npm ERR! peer **@angular/core@"^6.1.0 || ^7.0.0 || ^8.0.0 || ^9.0.0 || ^10.0.0 || ^11.0.0 || ^12.0.0"** from **apollo-angular**

npm ERR! node_modules/apollo-angular

npm ERR! **apollo-angular@"^2.6.0"** from the root project

npm ERR!

npm ERR! Conflicting peer dependency: **@angular/core@12.2.17**

npm ERR! node_modules/@angular/core

npm ERR! peer **@angular/core@"^6.1.0 || ^7.0.0 || ^8.0.0 || ^9.0.0 || ^10.0.0 || ^11.0.0 || ^12.0.0"** from **apollo-angular**

npm ERR! node_modules/apollo-angular

npm ERR! **apollo-angular@"^2.6.0"** from the root project

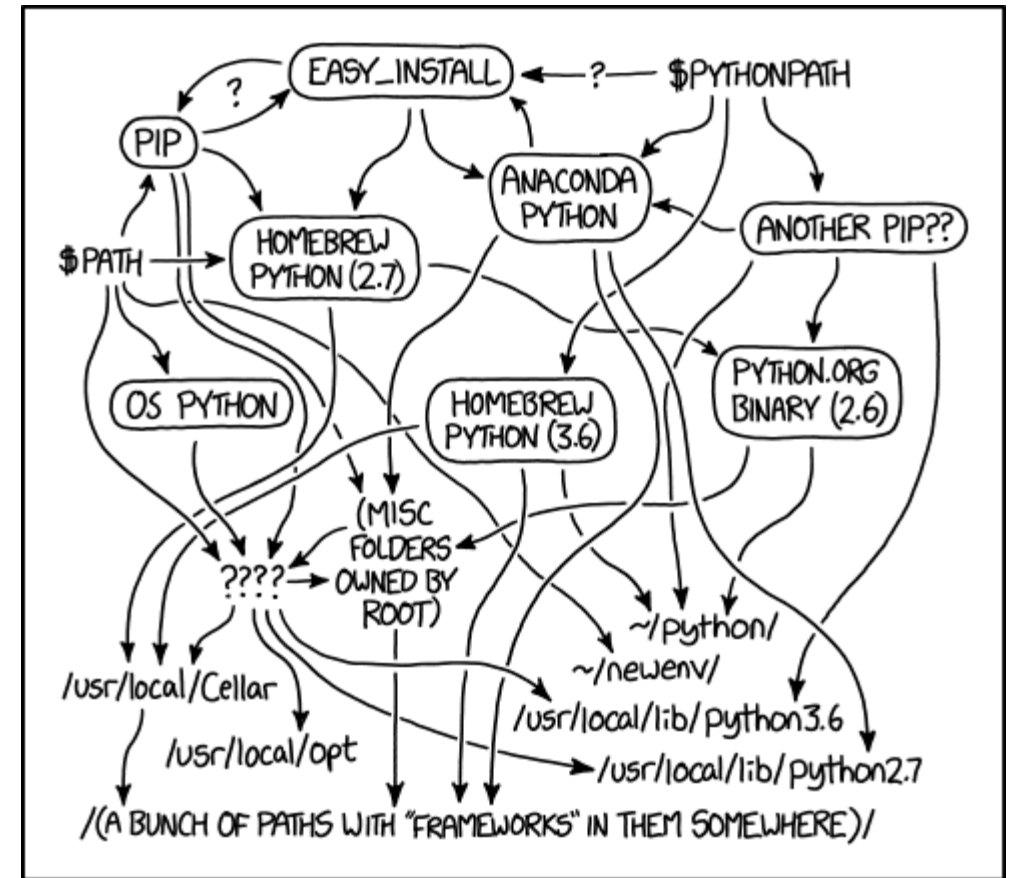
npm ERR!

Agenda

- The state of dependency hell
 - What is dependency hell
 - Why it is bad
- Navigating out of dependency Hell

Dependency hell == neglecting dependency management

- Characterised by
 - **outdated** dependencies
 - **unused** dependencies
 - A **fragile balance** between dependencies



MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED
THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

Neglecting dependencies lead to poorer dev-experience

CATcher: Node 14.21.2

20.11.1 LTS

Recommended For Most Users

We are here

Node.js Version	Codename	Release Date	npm
v21.7.1	-	2024-03-08	v10.5.0
v20.11.1	Iron	2024-02-13	v10.2.4
v19.9.0	-	2023-04-10	v9.6.3
v18.19.1	Hydrogen	2024-02-13	v10.2.4
v17.9.1	-	2022-06-01	v8.11.0
v16.20.2	Gallium	2023-08-08	v8.19.4
v15.14.0	-	2021-04-06	v7.7.6
v14.21.3	Fermium	2023-02-16	v6.14.18

Developing with outdated environments incurs tech debt

Developing feature with outdated dependency



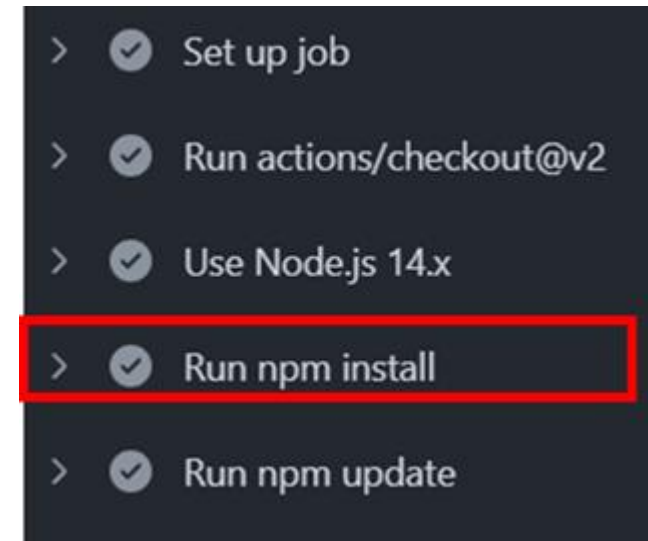
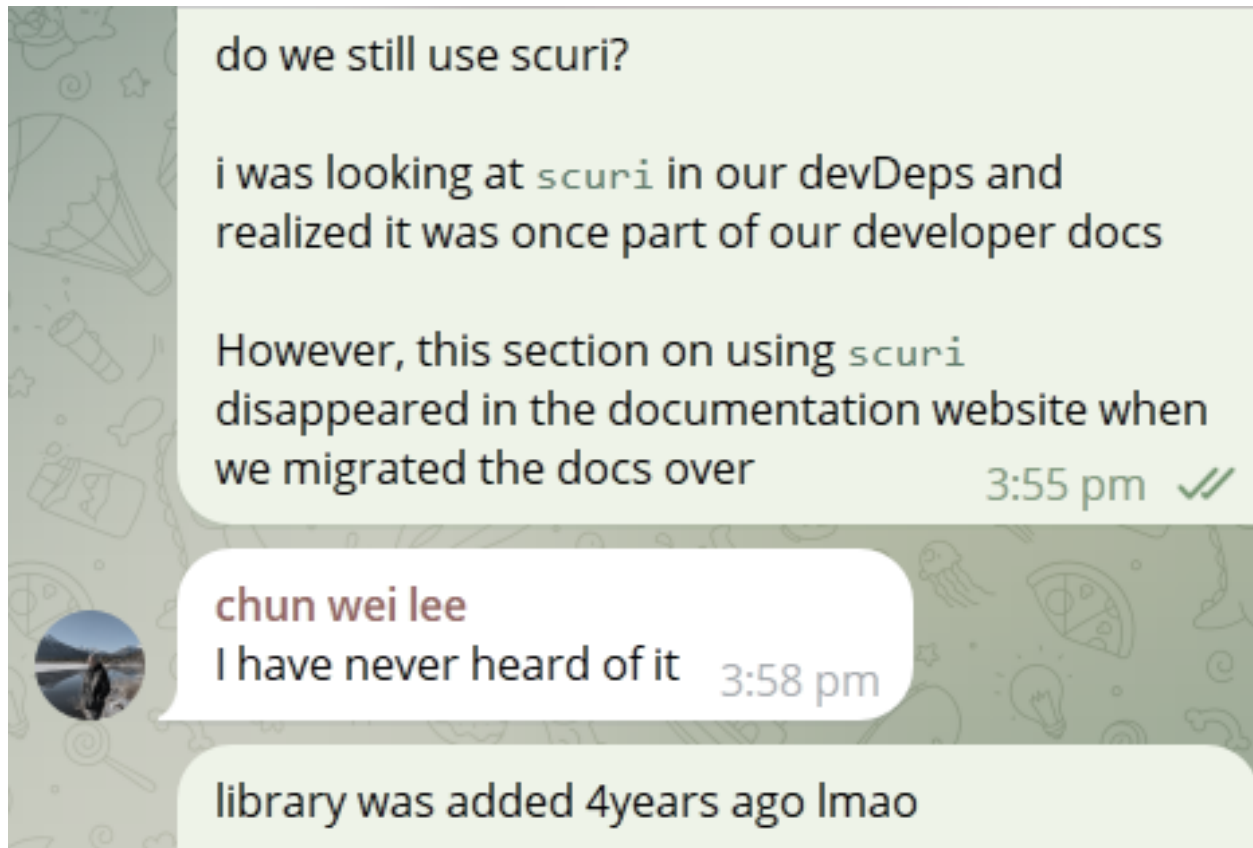
Dependency deprecates something in next version of dependency



Need to re-write feature when we upgrade dependency



Unused dependencies confuse your developers



They also slow down your workflows

Outdated dependencies are vulnerable

```
added 1945 packages from 1442 contributors and audited 1970 packages in 380.355s
```

```
176 packages are looking for funding  
  run `npm fund` for details
```





```
found 69 vulnerabilities (4 low, 46 moderate, 15 high, 4 critical)  
  run `npm audit fix` to fix them, or `npm audit` for details
```


Agenda

- The state of dependency hell
 - What is dependency hell
 - Why it is bad
- Navigating out of dependency Hell

Identify outdated and unused dependencies

- many node.js tools available for this
 - npm outdated
 - npm-check-update
 - depcheck
 - npm-check

ts-node	 NOTUSED?	Still using ts-node? Depcheck did not find code similar to <code>require('ts-node')</code> or <code>import from 'ts-node'</code> . Check your code before removing as depcheck isn't able to foresee all ways dependencies can be used. Use rc file options to remove unused check, but still monitor for outdated version: <code>.npmcheckrc {"depcheck": {"ignoreMatches": ["ts-node"]}}</code> Use <code>--skip-unused</code> to skip this check. To remove this package: <code>npm uninstall ts-node --save-dev</code>
typescript	 MAJOR UP	Major update available. https://www.typescriptlang.org/ <code>npm install typescript@5.4.2 --save-dev</code> to go from 4.3.5 to 5.4.2
wait-on	 MAJOR UP	Major update available. https://github.com/jeffbski/wait-on#readme <code>npm install wait-on@7.2.0 --save-dev</code> to go from 3.3.0 to 7.2.0
	 NOTUSED?	Still using wait-on? Depcheck did not find code similar to <code>require('wait-on')</code> or <code>import from 'wait-on'</code> . Check your code before removing as depcheck isn't able to foresee all ways dependencies can be used. Use rc file options to remove unused check, but still monitor for outdated version: <code>.npmcheckrc {"depcheck": {"ignoreMatches": ["wait-on"]}}</code> Use <code>--skip-unused</code> to skip this check. To remove this package: <code>npm uninstall wait-on --save-dev</code>

Identifying deprecated libraries

```
PS C:\Users\Chee Hong\Desktop\CS3281 - Thematic Systems Project I\CATcher> npm i
npm WARN deprecated tslint@6.1.3: TSLint has been deprecated in favor of ESLint. Please see https://github.com
npm WARN deprecated sourcemap-codec@1.4.8: Please use @jridgewell/sourcemap-codec instead
npm WARN deprecated @babel/plugin-proposal-async-generator-functions@7.14.7: This proposal has been merged to
d. Please use @babel/plugin-transform-async-generator-functions instead.
npm WARN deprecated core-js@3.16.0: core-js@<3.23.3 is no longer maintained and not recommended for usage due
tection in old core-js versions could cause a slowdown up to 100x even if nothing is polyfilled. Some versions
o the actual version of core-js.
npm WARN deprecated core-js@2.6.12: core-js@<3.23.3 is no longer maintained and not recommended for usage due
tection in old core-js versions could cause a slowdown up to 100x even if nothing is polyfilled. Some versions
o the actual version of core-js.
npm WARN deprecated @hapi/oi@15.1.1: Switch to 'npm install ioi'
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues
npm WARN deprecated @babel/plugin-proposal-class-static-block@7.21.0: This proposal has been merged to the ECM
se use @babel/plugin-transform-class-static-block instead.
npm WARN deprecated @babel/plugin-proposal-private-property-in-object@7.21.11: This proposal has been merged t
ned. Please use @babel/plugin-transform-private-property-in-object instead.
npm WARN deprecated @npmcli/move-file@1.1.2: This functionality has been moved to @npmcli/fs
npm WARN deprecated chokidar@2.1.8: Chokidar 2 does not receive security updates since 2019. Upgrade to chokid
npm WARN deprecated @hapi/address@2.1.4: Moved to 'npm install @sideway/address'
npm WARN deprecated @hapi/bourne@1.3.2: This version has been deprecated and is no longer supported or maintai
npm WARN deprecated @hapi/hoek@8.5.1: This version has been deprecated and is no longer supported or maintaine
npm WARN deprecated @hapi/topo@3.1.6: This version has been deprecated and is no longer supported or maintaine
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random()
tps://v8.dev/blog/math-random for details.
npm WARN deprecated chokidar@2.0.4: Chokidar 2 does not receive security updates since 2019. Upgrade to chokid
```

Investigate problematic dependencies

- After identifying the problematic dependencies, we can start by evaluating if anything else depends on it
- Using ``npm ls <dependency-name>``

```
PS C:\Users\Chee Hong\Desktop\CS3281 - Thematic Systems Project I\CATcher> npm ls request
CATcher@3.5.3 C:\Users\Chee Hong\Desktop\CS3281 - Thematic Systems Project I\CATcher
`-- wait-on@3.3.0
   `-- request@2.88.2
```

Evaluate problematic dependencies

- Check how the problematic dependency is used
 - Review the dependency's documentation
 - Do a search in your repository

```
wait-on 🤖 MAJOR UP Major update available
        😞 NOTUSED? Still using wait-on?
npm install wait-on@7.2
Depcheck did not find c
Check your code before
Use rc file options to
.npmcheckrc {"depch
```

Unused? + outdated

```
npmjs.com/package/wait-on#usage

Usage

Use from command line or using Node.js programmatic API.
```

Check npm docs

```
6 years ago Add project boilerplate code 77 "wait-on": "3.2.0",
```

Check git blame

```
"electron:serve-tsc": "tsc -p tsconfig-serve.json",
"electron:serve": "wait-on http-get://localhost:4200/ && npm run electron:serve-tsc && electron . --serve",
"electron:local": "npm run build:prod && electron .",
```

Aha! Turns out it was used as a CLI command in past version

There is no secret recipe

- Often a combination of tried-and-tested **techniques** such as
 - npm outdated / npm-check
 - npm ls <dependency-name>
 - Checking the dependency's README
 - Checking the dependency's CHANGELOG
 - Doing some manual investigative work

navigate out of dependency hell
and lead your projects to success
today!

