

# Update Dependencies

Class Note • 1 backlink • Tag

## 1. Update npm

```
npm install -g npm@10.9.0
```

```
(base) chris@YINGdeMacBook-Air codesnip % npm install -g npm@10.9.0
removed 25 packages, and changed 97 packages in 2s

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

## 2. Run ncu to check available update and ncu -i to update

```
(base) chris@YINGdeMacBook-Air codesnip % ncu
Checking /Users/chris/UCSDClass/24Fall/CSE210/codesnip/package.json
[=====] 18/18 100%

@types/glob      ^7.1.3  -> ^8.1.0
@types/mocha     ^8.0.0  -> ^10.0.10
@types/mustache  ^4.0.1  -> ^4.2.5
@types/node      ^12.11.7 -> ^22.9.1
@types/vscode    ^1.75.0  -> ^1.95.0
@typescript-eslint/eslint-plugin ^4.1.1  -> ^8.15.0
@typescript-eslint/parser ^4.1.1  -> ^8.15.0
eslint          ^7.9.0  -> ^9.15.0
glob            ^7.1.6  -> ^11.0.0
mocha           ^9.1.3  -> ^10.8.2
mustache        ^4.0.1  -> ^4.2.0
ts-loader       8.2.0   -> 9.5.1
typescript      ^4.0.2  -> ^5.6.3
vscode-test     ^1.4.0  -> ^1.6.1
webpack         ^5.76.0 -> ^5.96.1
webpack-cli     ^4.2.0  -> ^5.1.4
y18n            >=4.0.1 -> >=5.0.8
```

```
(base) chris@YINGdeMacBook-Air codesnip % ncu -u
Upgrading /Users/chris/UCSDClass/24Fall/CSE210/codesnip/package.json
[=====] 18/18 100%

@types/glob      ^7.1.3  -> ^8.1.0
@types/mocha     ^8.0.0  -> ^10.0.10
@types/mustache  ^4.0.1  -> ^4.2.5
@types/node      ^12.11.7 -> ^22.9.1
@types/vscode    ^1.75.0  -> ^1.95.0
@typescript-eslint/eslint-plugin ^4.1.1  -> ^8.15.0
@typescript-eslint/parser ^4.1.1  -> ^8.15.0
eslint          ^7.9.0  -> ^9.15.0
glob            ^7.1.6  -> ^11.0.0
mocha           ^9.1.3  -> ^10.8.2
mustache        ^4.0.1  -> ^4.2.0
ts-loader       8.2.0   -> 9.5.1
typescript      ^4.0.2  -> ^5.6.3
vscode-test     ^1.4.0  -> ^1.6.1
webpack         ^5.76.0 -> ^5.96.1
webpack-cli     ^4.2.0  -> ^5.1.4
y18n            >=4.0.1 -> >=5.0.8
```

## 3. Run npm install to install the dependencies

- There are some warning, we can ignore them.

```
(base) chris@YINGdeMacBook-Air codesnip % npm install
npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerful.
npm warn deprecated rimraf@2.7.1: Rimraf versions prior to v4 are no longer supported
npm warn deprecated rimraf@3.0.2: Rimraf versions prior to v4 are no longer supported
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated fstream@1.0.12: This package is no longer supported.
npm warn deprecated vscode-test@1.6.1: This package has been renamed to @vscode/test-electron, please update to the new name
npm warn deprecated glob@8.1.0: Glob versions prior to v9 are no longer supported

added 347 packages, and audited 348 packages in 23s

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

found 0 vulnerabilities
```

4. Run `npm run compile` to try to compile the files

- There are errors in the
  1. `src/data/fileDataAccess.ts`
  2. `src/test/suite/index.ts`
- Fix the errors
  1. In `src/data/fileDataAccess.ts`, the `fs.readFileSync(this._dataFile, this._encoding);` takes a `BufferEncoding` as the encoding type instead of a `string`

TypeScript ▾

```
//private _encoding = 'utf8';  
    private _encoding: BufferEncoding = "utf-  
8"; // fs.readFileSync() takes BufferEncoding  
instead of string
```

2. In `index.ts`, the way to import `glob` has changed.

### Correct Import for glob (with Promise support)

If you are using `glob` and want to work with the **Promise-based version** of the function, you need to make sure you're importing the correct version of `glob`.

Here's how you can import and use it correctly, assuming you're using the newer version ( `glob 8.x` or higher), which supports promises:

TypeScript ▾

```
// import * as glob from 'glob'; // For glob  
7.x and below  
import glob from 'glob';
```

This is because `glob` has moved to exporting the function as the default export in the newer versions. In older versions, you might have been using `import * as glob from 'glob'`, but now it's the default export.

### Using `glob` with Promises

With `glob 8.x+` and the correct import, you can use it with Promises as follows (glob 7.x and below is call-back only):

```
TypeScript ∨  
  
export function run(): Promise<void> {  
  // Create the mocha test  
  const mocha = new Mocha({  
    ui: 'tdd',  
    color: true  
  });  
  
  const testsRoot = path.resolve(__dirname,  
    '..');  
  
  return new Promise((c, e) => {  
    // Call-back version  
    // glob('**/*.test.js', { cwd:  
testsRoot }, (err, files) => {  
      //      if (err) {  
      //          return e(err);  
      //      }  
  
      // Promise version  
      glob('**/*.test.js', { cwd: testsRoot  
    })  
  
      .then((files: string[]) => {  
        // other stuff are the same...  
  
      });  
    });  
  });  
}
```

- ESLint
  - The new ESLint configuration format introduced in ESLint v8.21.0. In this format, `eslint.config.js` and certain command-line flags, such as `--ext`, are no longer supported. And we need to migrate the `.eslintrc.json` file to `eslint.config.js` using:

```
TypeScript ∨  
  
// https://eslint.org/docs/latest/use/configure/migration-  
guide#migrate-your-config-file  
npx @eslint/migrate-config .eslintrc.json
```

- We need to change the ESLint command for npm

TypeScript ▾

```
// package.json
// "lint": "eslint src --ext ts",
"lint": "eslint src",
```

- And change the test configure for GitHub test

TypeScript ▾

```
// .github/workflows/unit-tests.yml
// node-version: 16.x
node-version: 22.x
```

TypeScript ▾

```
// .github/workflows/code-analysis.yml
- name: Run ESLint
  // run: npx eslint src --ext ts --max-warnings=0
  run: npx eslint src --max-warnings=0
```