# package-json-from-dist

Sometimes you want to load the package.json into your TypeScript program, and it's tempting to just import '../package.json', since that seems to work.

However, this requires tsc to make an entire copy of your package.json file into the dist folder, which is a problem if you're using something like [tshy](https://github.com/isaacs/tshy), which uses the package.json file in dist for another purpose. Even when that does work, it's asking the module system to do a bunch of extra fs system calls, just to load a version number or something. (See [this issue](https://github.com/isaacs/tshy/issues/61).)

This module helps by just finding the package.json file appropriately, and reading and parsing it in the most normal fashion.

## Caveats

This *only* works if your code builds into a target folder called dist, which is in the root of the package. It also requires that you do not have a folder named node\_modules anywhere within your dev environment, or else it'll get the wrong answers there. (But, at least, that'll be in dev, so you're pretty likely to notice.)

If you build to some other location, then you'll need a different approach. (Feel free to fork this module and make it your own, or just put the code right inline, there's not much of it.)

## USAGE

// src/index.ts

import {

findPackageJson,

loadPackageJson,

} from 'package-json-from-dist'

const pj = findPackageJson(import.meta.url)

console.log(`package.json found at ${pj}`)

const pkg = loadPackageJson(import.meta.url)

console.log(`Hello from ${pkg.name}@${pkg.version}`)

If your module is not directly in the ./src folder, then you need to specify the path that you would expect to find the package.json when it's *not* built to the dist folder.

// src/components/something.ts

import {

findPackageJson,

loadPackageJson,

} from 'package-json-from-dist'

const pj = findPackageJson(import.meta.url, '../../package.json')

console.log(`package.json found at ${pj}`)

const pkg = loadPackageJson(import.meta.url, '../../package.json')

console.log(`Hello from ${pkg.name}@${pkg.version}`)

When running from CommmonJS, use \_\_filename instead of import.meta.url.

// src/index.cts

import {

findPackageJson,

loadPackageJson,

} from 'package-json-from-dist'

const pj = findPackageJson(\_\_filename)

console.log(`package.json found at ${pj}`)

const pkg = loadPackageJson(\_\_filename)

console.log(`Hello from ${pkg.name}@${pkg.version}`)

Since [tshy](https://github.com/isaacs/tshy) builds *both* CommonJS and ESM by default, you may find that you need a CommonJS override and some //@ts-ignore magic to make it work.

src/pkg.ts:

import {

findPackageJson,

loadPackageJson,

} from 'package-json-from-dist'

//@ts-ignore

export const pkg = loadPackageJson(import.meta.url)

//@ts-ignore

export const pj = findPackageJson(import.meta.url)

src/pkg-cjs.cts:

import {

findPackageJson,

loadPackageJson,

} from 'package-json-from-dist'

export const pkg = loadPackageJson(\_\_filename)

export const pj = findPackageJson(\_\_filename)