

Global & Local npm Packages

In the last lecture, we added `nodemon` as a local dependency to our project.

The good thing about local dependencies is that you can share projects **without the `node_modules`** folder (where they are stored) and you can run `npm install` in a project to then re-create that `node_modules` folder. This allows you to share only your source code, hence reducing the size of the shared project vastly.

The attached course code snippets also are shared in that way, hence you need to run `npm install` in the extracted packages to be able to run my code!

I showed that `nodemon app.js` would **not work** in the terminal or command line because we don't use local dependencies there but global packages.

You could install `nodemon` globally if you wanted (this is NOT required though - because we can just run it locally): `npm install -g nodemon` would do the trick. Specifically the `-g` flag ensures that the package gets added as a global package which you now can use anywhere on your machine, directly from inside the terminal or command prompt.