In the upcoming lectures, I will be using the **Express** framework in Node. I am doing this because Express makes it easy to define routes.

This is beyond the scope of the course, so if you don't want to code alongside me, that is perfectly fine. You can sit back, relax, and watch.

But if you are keen to learn about server-side code (at a high level), then read on…

There are two ways you can install Express.

(Note: I have assumed you have installed Node on your machine. Node comes shipped with NPM automatically.)

#1. Easiest way: Install Express globally (this is fine for this course)

Go to your terminal and type the following:

```
npm install express -g
```

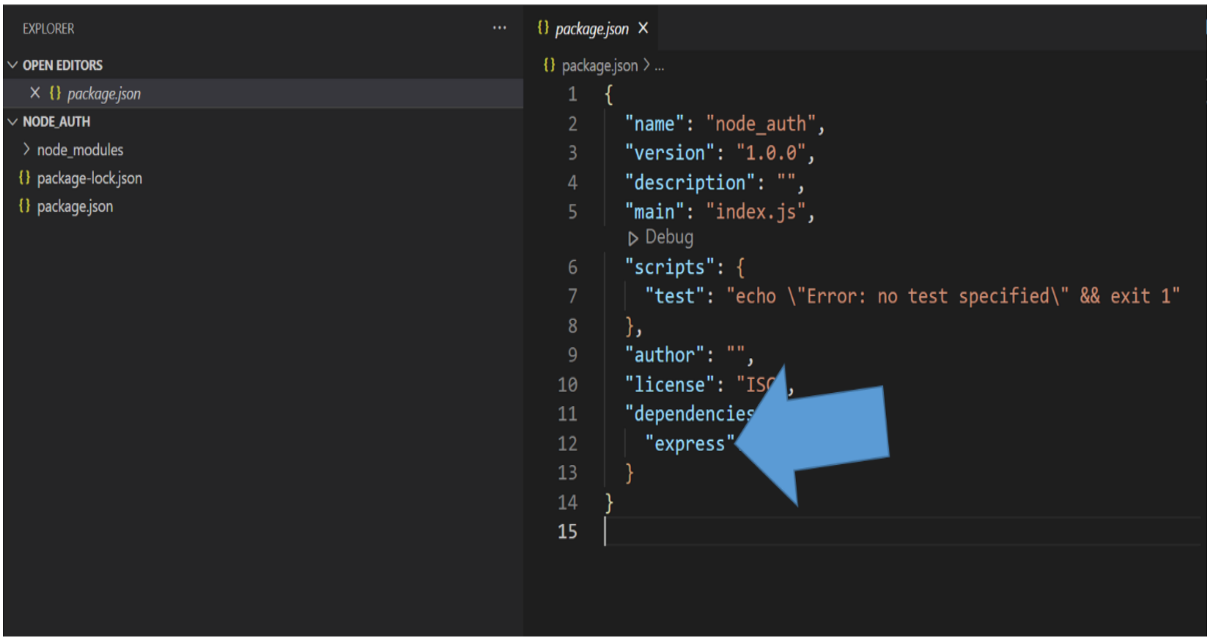
This method installs the package globally on your machine. Installing it in this way will allow you to use the `express` command without having to install Express on every project.

#2. Locally to your project folder

Make sure you navigate to the folder you are writing your code in.

You have to set up a Node project by typing `npm init`. This will set up a **package.json** file for you in your current directory.

Next, all you have to do is install express by typing `npm install express`. This will add a **node_modules** folder to your current folder, as follows:



```
{
  "name": "node_auth",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies
    "express"
  }
}

```

Note: In the real world, you will likely have Express installed in your local project directory (which will ensure that your app is using the latest Express version), **AND** you will have it installed globally so when you develop code, it's quick to access. As you will notice in this particular section, I don't install Express locally. I just use it directly and Node knows where to find it.

Later on in this course, when we are building our main course project, I will install a few modules locally to our project (Express, Nodemon, and even a module called cors).