Checkpoint 6

The package.json file

Each folder that you create and plan to run with Node can be considered a Node package or project. These projects can contain a configuration file that gives additional information about the project. The configuration file also allows for some interesting functionality through the use of npm, the command-line tool that you installed at the beginning of this module.

Outline

In this checkpoint, you will learn about this configuration file: the `package.json` file. You will learn how to create it and use it to run scripts from the command line with npm. Learning this will increase your efficiency as a developer and prepare you for installing additional packages from the web.

By the end of this checkpoint, you will be able to do the following:

- Create and use the `package.json` file to run scripts

Configuration files

So far, the files that you've created have been JavaScript files that you've been able to run with Node. These files are, obviously, critical to running code in that they have code in them. As you advance throughout this program, you will also come across a number of configuration files, which are essential to larger projects. *Configuration files* can generally be described as any file that's necessary to run an overall project but may not have explicit code (functions) inside of it.

For example, your text editor (whether it's Visual Studio Code, Sublime, or something else) likely has a settings file that defines how your text editor works. These settings may determine how your code is spaced or what colors your text editor shows. These settings don't explicitly run your text editor, but they do shape *how* the text editor is run.

Outline

The `package.json` file

For Node projects, the main configuration file is the `package.json` file. This file contains meta information about the project and allows you to customize some commands that can be run.

The `package.json` file can be created through npm. After navigating to the folder where you want to create your project, you can create a `package.json` file by running the following command on the command line:

```
npm init -y
```

The above command will create a new `package.json` file in t directory where it was run. It will also fill in some default values.

Note: Each Node project should only have a single `package.json` file.

Do this

Create a `package.json` file

Create a new directory called `first-package-json` on your computer. On your command line, navigate to that directory and run the above command to create a `package.json` file.

Take a look at the file that was just created. Note that it looks similar to a JavaScript object.

JSON

Outline

The file that you created isn't a JavaScript file—it's a JSON file. JSON stands for *JavaScript Object Notation*, which is *different* from JavaScript. JSON looks like a JavaScript object but has more limitations. In particular, when writing JSON, you must consider the following:

- All keys in a JSON file must be wrapped in quotations.
- A JSON file cannot include JavaScript comments or variables. Instead, the file must only contain a single JavaScript object.

Values such as booleans, numbers, strings, arrays, and other objects are valid in JSON.

Keys

If you used the command suggested earlier in this checkpoint, `package.json` file will look something like this by default:

```
{
  "name": "first-package-json",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exi
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

All of the above keys are optional. There are a few that you will want to keep—especially the following:

- `name`: This is the name of your package. By default, it will take the name from your folder's name.
- `description`: This is a description of your project. You can place whatever you want here.
- `scripts`: This is where you will set up scripts to be run through npm. You will learn more about scripts below.

You can learn more about the `package.json` file and other keys in the package.json documentation.

Scripts

Right now, one of the most useful skills that you can learn is how to build scripts with the `package.json` file. The `scripts` key has a value that is an object. Each key in that object is the name of a script that can be run from the command line. The value associated with that

key is a string containing any valid command-line code. Running that script with npm will execute the code on the command line.

For example, take the following default script:

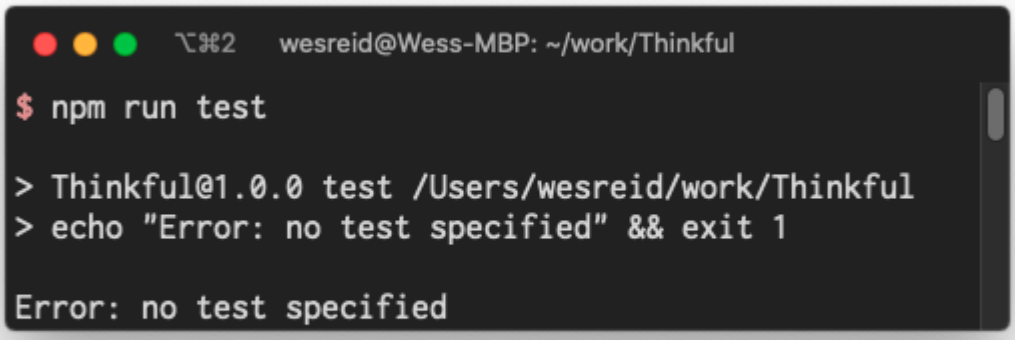
```
{
  "test": "echo \"Error: no test specified\" && exit
}
```

This script can be run with the following command:

```
npm run test
```

Outline

When the above command is run on the command line in the folder where the `package.json` file exists, it will run the value as a script. Here's an example:



After `npm run test` was run, the value of the `test` command was run. That is, a specific message was echoed.

You can create new scripts by adding new key-value pairs. For example, the following script will simply run the `pwd` command on the command line.

```
"print-working-directory": "pwd"
```

To run the command, you would type the following:

```
npm run print-working-directory
```

Note: If you try to include the above script, make sure to use the correct syntax for your JSON file.

Outline

Do this

Run JavaScript with npm

In the `first-package-json` folder, create a new file called `main.js`. Add the following code to it:

```
console.log("Hello, world!");
```

Then, add a script called `dev` that will run the following command:

```
node main.js
```

Afterward, on the command line, you should be able to run the following command and see `"Hello, world!"` printed to the console.

```
npm run dev
```

Default commands

The npm program expects you to use some common terms like `start` and `test`. For these common terms, you can skip the `run` part of the command.

```
npm test
```

However, you will still need to use the `run` part of the command for custom scripts.

Outline

```
npm run dev
```

More of the supported scripts can be found in the npm scripts documentation.

Checkpoint

This checkpoint will be autograded. Please click the link below to open your assignment in a new tab. Once you complete the assignment, you will see a button allowing you to submit your answers and move on to the next checkpoint.

Your work

03.18.21 **Approved**

Completed

Next checkpoint

How would you rate this content?

Report a typo or other issue

Outline

Go to Overview