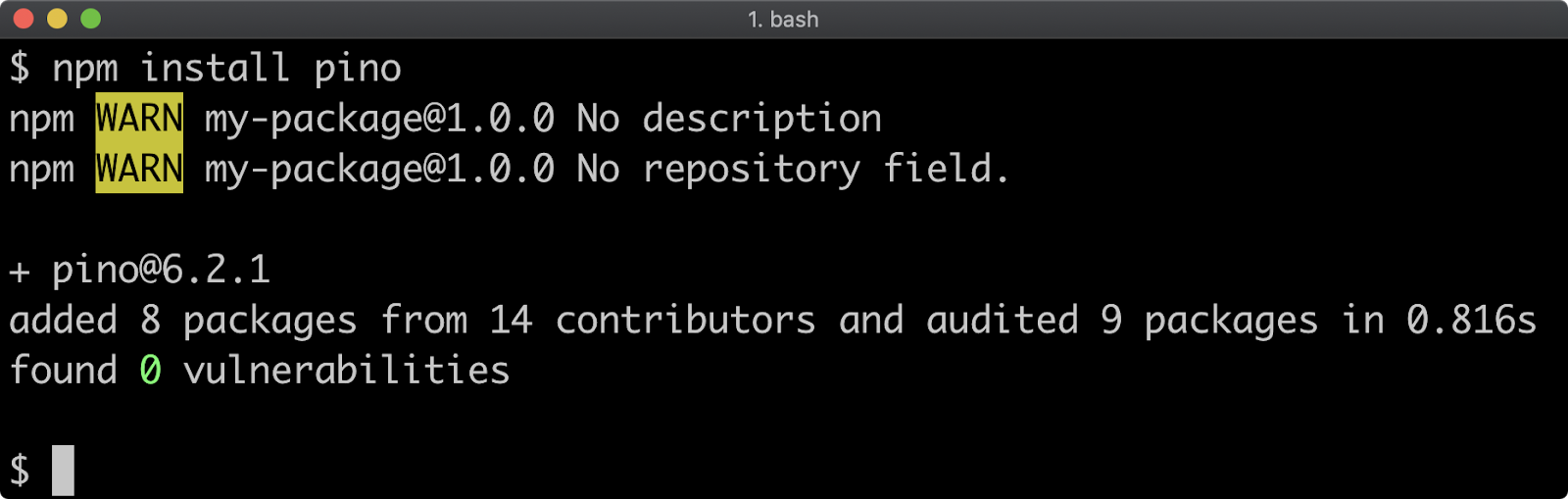
**Installing Dependencies**

Once a folder has a **package.json** file, dependencies can be installed.

Let's install a logger:

**npm install pino**



Information about any ecosystem package can be found on [npmjs.com](https://www.npmjs.com/), for instance for information about the logger we installed see [Pino's Documentation](https://www.npmjs.com/package/pino" \t "_blank).

Once the dependency is installed the **package.json** file will have the following content:

**{  
  "name": "my-package",  
  "version": "1.0.0",  
  "description": "",  
  "main": "index.js",  
  "scripts": {  
    "test": "echo \"Error: no test specified\" && exit 1"  
  },  
  "keywords": [],  
  "author": "",  
  "license": "ISC",  
  "dependencies": {  
    "pino": "^6.2.1"  
  }  
}**

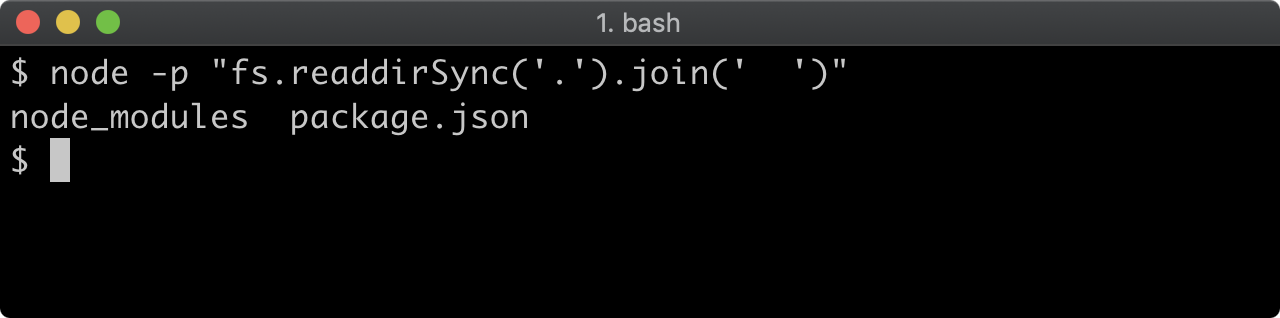
Running the **npm install** command has modified the **package.json** file by adding a **"dependencies"** field:

**"dependencies": {  
    "pino": "^6.2.1"  
  }**

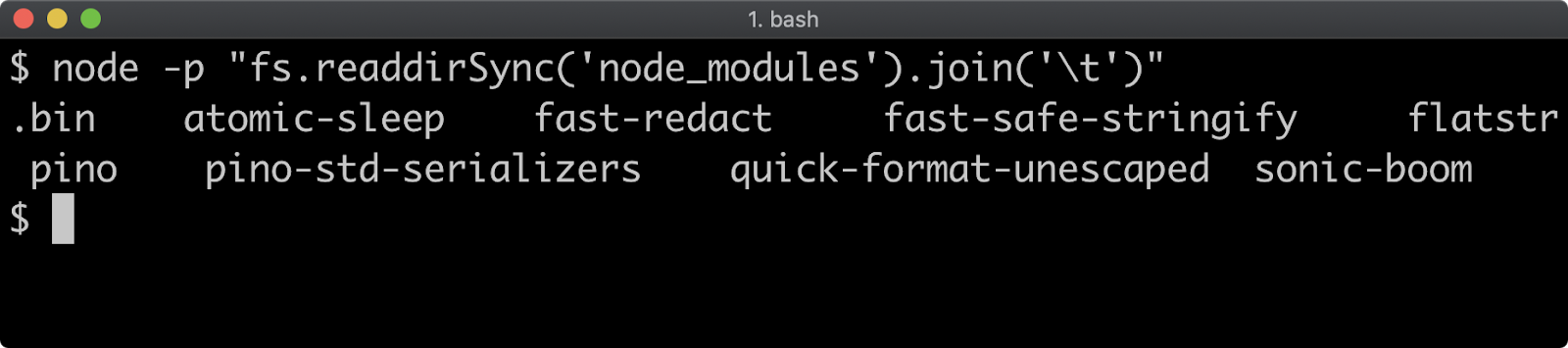
The **"dependencies"** field contains an object, the keys of the object contain dependency namespaces, the values in the object contain the Semver range version number for that dependency. We will explore the Semver format later in this section.

Running **npm install pino** without specifying a version will install the latest version of the package, so the version number may vary when following these steps.

In addition, a **node\_modules** folder will have been added into the **my-package** folder:



The **node\_modules** folder contains the logger package, along with all the packages in its dependency tree:



The **npm install** command uses a maximally flat strategy where all packages in a dependency tree placed at the top level of the **node\_modules** folder unless there are two different versions of the same package in the dependency tree, in which case the packages may be stored in a nested **node\_modules** folder.

The **npm ls** command can be used to describe the dependency tree of a package:



Notice how the **flatstr** sub-dependency occurs twice in the output. The second occurrence has the word **deduped** next to it. The **flatstr** module is a dependency of both **pino**and its direct dependency **sonic-boom**,but both**pino**and**sonic-boom**rely on the same version of**flatstr.**Which allows npm to simply place a single**flatstr**package in the**node\_modules**folder**.**

*Cont'd on the next page.*