Node has several built-in modules, but they’re rarely enough; third-party packages are indispensable when making applications.

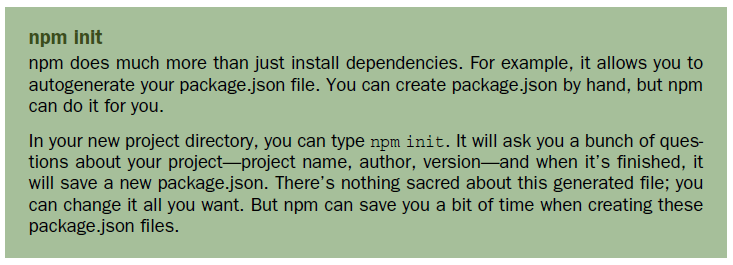
Package.json:

Every Node project sits in a folder, and at the root of every Node project there’s a file called package.json.

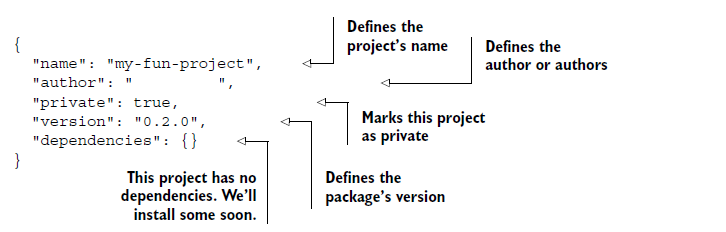
“package dot json” is a pretty simple JSON file that defines project metadata like

the name of the project, its version, and its authors. It also defines the project’s dependencies.

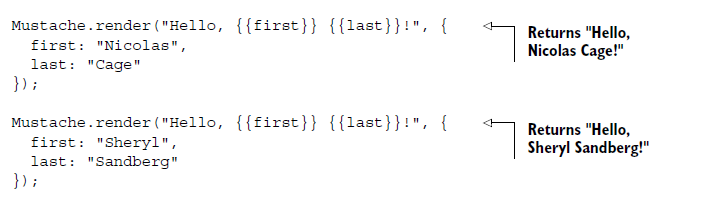
Example 1: init a project and check its package.json



A simple package.json file:



Example 2: use Mustache templating system:

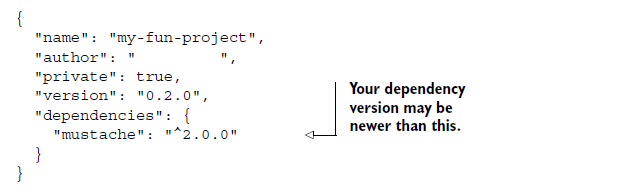


From the root of this directory, run npm install mustache --save. (You must run this command from the root of this directory so that npm knows where to put things.)

1. This command creates a folder in this directory called node\_modules.
2. Then it downloads the latest version of the Mustache package and puts it into this new

node\_modules folder (look inside to check it out). The --save flag adds it to your package.json.

Your package.json file should look similar to the one in the next listing, but it will now have the latest version of the Mustache package.



If you left off the --save flag, you’d see the new node\_modules folder and it would

have Mustache inside, but nothing would be present in your package.json.

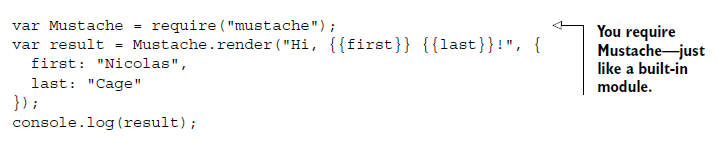
The reason you want dependencies listed in your package.json is so that someone else can install

the dependencies later if you give them the project—they need only to run npm

install with no arguments. Node projects typically have dependencies listed in their

package.json but they don’t come with the actual dependency files (they don’t include the node\_modules folder).

Using the mustache module:



Run it:

node mustache-test.js