**How to automatically generate an Angular unit test**

When you use the Angular CLI to scaffold a project, it generates unit test specs for the default app component. Following the initial project setup, we’ll need to write test specifications for each component, directive, and more. In most cases, this will be time-consuming because we will have to write these specifications from the ground up. Fortunately, the Angular ecosystem created the ngentest package to automate the generation of test specs for each component, directive, and others.

Assuming you wrote the code for your component and want to write a test for it, you’ll have to install the ngentest package:

$ npm install ngentest -g

Next, you’ll run the following command to auto-generate the unit test specs for your component:

$ gentest component-name.ts

We can also auto-generate the unit test specs for directives, pipes, and services:

$ gentest directive-name.ts -s # output to directive-name.spec.ts

$ gentest pipe-name.ts # output to pipe-name.test.ts

$ gentest service-name.ts. # output to service-name.test.ts

**How ngentest works**

ngentest parses the file name next to the gentest command and determines the proper file type. In our case, it is component-name.ts as seen in our previous command.

Next, ngentest builds data for a unit test from the parsed TypeScript using the contents of the file, such as:

* Class
* Imports
* List of functions to test
* Inputs and outputs
* Mocks

Finally, ngentest generates the unit test.

The gentest package does not generate 100% test coverage, so you’ll need to modify the generated unit test specs in order to achieve that