webpack Basics with Angular

webpack is a static module bundler for modern JavaScript applications

**In webpack you configure the following:**

Entry: the module where webpack should start

Output: where webpack emits the bundles it creates

Loaders: enable webpack to process more than just JavaScript files

Plugins: perform a wide range of tasks like minification, for example

You set all these up in the webpack configuration file which is typically named: webpack.config.js

So Where is the webpack.config.js?

“But wait Todd, I just searched my entire Angular project and there is no webpack.config.js. Are you sure Angular CLI still uses webpack?”

OK, you’re right, there is no webpack.config.js in our Angular project. And, some healthy skepticism is a good thing. But, let’s take a quick look and verify that Angular does in fact depend on webpack.

In our project we can find a local copy of the Angular CLI here: node\_modules\@angular\cli

In that directory you should see the package.json file for Angular CLI. Open it and take a look at the dependencies and among others you will see webpack, webpack-dev-server, etc. For example:

Also, in node\_modules you can see that these webpack based dependencies actually did get installed. For example: node\_modules/webpack

So, we see that Angular CLI is, in fact, using webpack but it is only a black box to us. This is great if you aren’t a webpack expert. But, if you have taken the time to level-up your webpack skills, this might be unacceptable to you. Fear not, the Angular CLI has a solution for us.

Accessing webpack Config With ng eject

Once we have used Angular CLI to generate our seed application, we can switch to a native webpack approach using: ng eject

In the root of our application run:

ng eject

This will do the following:

Generate a webpack.config.js file in the root of our project based on the current build.

Sets the ejected flag to true in .angular-cli.json.

Updates the scripts in package.json to run based on webpack rather than Angular CLI.

Running After Eject

You should do an npm install after ejecting:

npm install

Now if we want to see our application running we can’t use ng serve anymore. We need to use:

npm run start

If we look in our package.json we can see that this script actually translates to:

webpack-dev-server --port=4200

So we can still go to http:\\localhost:4200 to see our application running in the Development server.

If we try to use ng serve or ng build or anything ng project specific, we will get an error:

ng build

An ejected project cannot use the build command anymore.

To create a full angular project and webpack manually: https://www.tektutorialshub.com/angular/angular-webpack-tutorial/