WEBPACK

Webpack is an open source JavaScript module bundler. It is used for bundling process of all the files into a single file at the destination folder.Its main purpose is to bundle JavaScript files for usage in a browser and also capable of transforming or packaging just about any resource or assets.

it can transform front-end assets like HTML, CSS and also even images.

Webpack is also used to configure both development mode and production mode.

Eg:- webpack.config.common.js

The above command is an configuration used for both development and production.

**webpack.config.dev.js** is our webpack configuration that we will use for development mode only.

**webpack.config.prod.js** is our webpack configuration that we will use for production mode only.

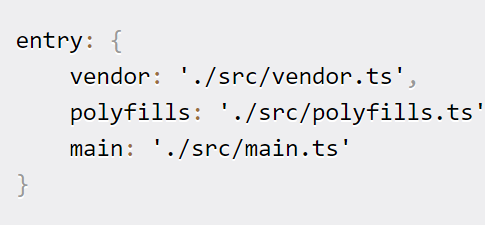
We have to configure some core concepts inorder to tell webpack how to bundle our application.

They are:

* Entry
* Output
* Loader
* Plugins

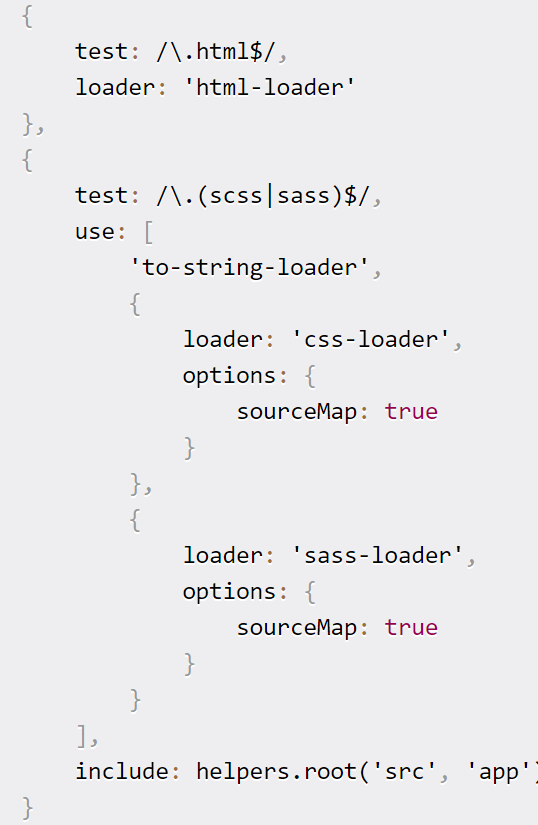
**Entry:**

For this application (and for most of them actually) we have 3 different entry points : **vendor.ts,polyfills.ts**and **main.ts.**



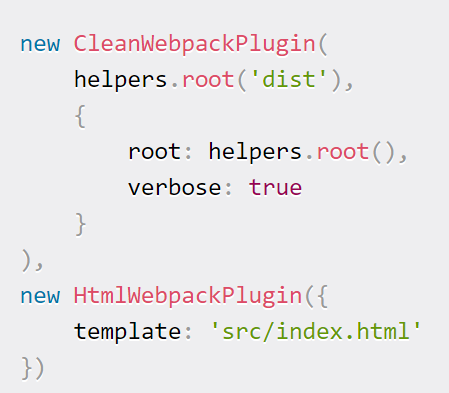
**Loader:**

Loading .scss files is a bit tricky for an Angular app.We load **.html** files with **html-loader**which is pretty standard.First of all, we must load sass files by using two loaders **sass-loader**and **css-loader.** If you want to make debugging easy, it’s really important to add **sourceMap** as **true in** options.



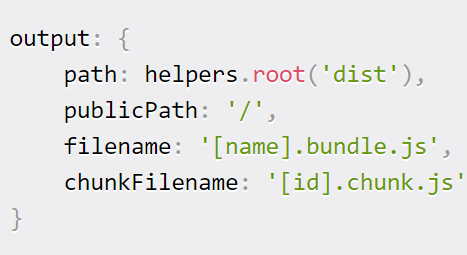
**Plugins:**

**CleanWebpackPlugin** will remove/clean your build folders before building again. **HtmlWebpackPlugin** will generate an HTML5 file for you that includes all your webpack bundles in the body using script tags.



**Output:**

The output key contains a set of options which instructs to webpack on how and where it should output your bundles, assets.Here we tell webpack to output our bundles to the dist folder.



NG-BUILD

The ng-build is a command which is used to build a project of type application or library,So that we can expect the final output as a library or application.This command must be executed from within a workspace directory. When we use this command to build a library a different builder is invoked which uses only 3 options to apply.They are:- ts-config,configuration and watch options.

In ng-build we have 2 modes

* Development Mode
* Production Mode

**Development Mode:**

The **ng-build** by default is in a development mode and this mode compiles the application into an output directory.

Eg : ng build

**Production Mode:**

Here we can run this mode by passing **–prod** to command.

Eg : ng build --prod

In production mode,we will be having the compressed files which are the same types from which they are originally compressed. i.e., the .html files are compressed to .html files only.

But in development mode all the files are converted to .js format.So inorder to convert the files to the same format after compressing is done.We can use the command – ec with true value in development mode.

Eg: ng build -- dev - ec true

In production mode we can reduce the final bundle.js file by following methods.

* Minification
* Tree Shaking
* Uglification

**Minification:**

The bundle contains a bunch of useless content such as:

* Unused functions, variables…
* A lot of whitespace.
* Comments.
* Non-mangled variables.

In order to reduce the size of the bundle we can use the minify script

**Uglification:**

It rewrites code to use short, cryptic variable and function names.

**TreeShaking:**

Tree shaking is the ability to remove any code that we are not actually using in our application from the final bundle. It's one of the most effective techniques to reduce the footprint of an application.

Run Angular on Node Server:

We should follow 3 steps inorder to run an angular project on node server.

1. After creating the complete angular project, by using **ng-build** command,build the project inorder to compress the files which are created at dist folder. Here all files are converted to .js format.
2. Now copy the compressed folder with the title name to the node project folder.We need to just add one middleware,So that angular project would be runned on node server port.
3. app.use(express.static('Table'));
4. Now run the node sever and then you can see your angular project running on node server with 8082 port.

