Laravel Mix

Npm build that helps in to compile all production assets(css and js) into a single file.

Normally during development, various css files are created and also js files, e.g components files in vue.

Hence theirs an need to compile all the different files into one file(often compressed) for production.

Laravel Mix is thus a useful tool to carry out the compiling.

NOTE: The standalone version of mix is used

Prerequisite of Installation

Before installation there are prerequisite softwares; npm and node

To check availability of both, run

```
npm -v
```

```
node -v
```

npm is the package installer for the installation of the mix

Steps for Installation of Laravel Mix

1. Make a directory for the project and cd into it, run

```
mkdir my-app && cd my-app
```

2. Next we create the package.json file, to add dependencies to it, run

```
npm init -y
```

3. To install laravel mix as a dependency into our project, run

```
npm install laravel-mix --save-dev
```

4. cross-env dependency is also added, run

```
npm install cross-env --save-dev
```

It tell npm what environment we are in, either production or development

So far in our directory a nodes-module directory is added to our project.

5. Import weback configuration to our root folder from the nodes-modules directory, run

```
cp node_modules/laravel-mix/setup/webpack.mix.js ./
```

6. Now that the mix is configured, make the following files for app.js, and app.scss into an ./src folder

   run

   ```
   mkdir src && touch src/app.{js,scss}
   ```

   The development is to be carried out in the two folders, which will later be compiled into a ./dist folder.

   The snippet below shows the webpack.mix.js file

   ```
   mix.js('src/app.js', 'dist/').sass('src/app.scss', 'dist/');
   ```

7. Next vue is to be installed so as, we are able to compile our own vue and not use the cdn, in the my-app/src directory run

   ```
   npm install vue --save-dev
   ```

8. Add npm scripts to the package.json files, to speed up the workflow. the scripts are available in laravel mix page.

   ```
   "scripts": {
       "dev": "npm run development",
       "development": "cross-env NODE_ENV=development
   node_modules/webpack/bin/webpack.js --progress --hide-modules --
   config=node_modules/laravel-mix/setup/webpack.config.js",
       "watch": "npm run development -- --watch",
       "hot": "cross-env NODE_ENV=development node_modules/webpack-dev-
   server/bin/webpack-dev-server.js --inline --hot --
   config=node_modules/laravel-mix/setup/webpack.config.js",
       "prod": "npm run production",
       "production": "cross-env NODE_ENV=production
   node_modules/webpack/bin/webpack.js --no-progress --hide-modules --
   config=node_modules/laravel-mix/setup/webpack.config.js"
   }
   ```

9. To run mix for the fisrt time, in the root directory, run

   ```
   npm run watch
   ```

   This compiles all the files, and installs any addition dependencies,

   Incase of error, run

   ```
   npm install

   npm run watch
   ```

   After successful installation, a dist directory is added into our root folder.

NOTE;

To compile the code for development we use,

```
npm run watch
```

But to compile for production

```
npm run prod
```

Usage

The app.css and app.js in the dist directory are the final compiled js and css for the project,

In the html markup, we pull in this files instead of the cdn, as shown

```
<script src="./dist/app.js"></script>
```

For the js, the app.js file in the src is used to pull in all the other js files for compling.

An example for using laravel mix

Html

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
    </head>

    <body>
        <div id="app">
           <list></list>
        </div>

        <script src="./dist/app.js"></script>
    </body>

</html>
```

my-app/src/app.js

```
import Vue from 'vue';
import Accordian from './components/Accordian.vue';
import List from "./components/List.vue";


new Vue ({
    el: "#app",
    components:{
        Accordian,
        List,
    },

    data: {
        items:[
            {id:"1", title:'Tiltle 1', description:'Description for tab 1' },
            {id:"2", title:'Tiltle 2', description:'Description for tab 2' },
            {id:"3", title:'Tiltle 3', description:'Description for tab 3' },
        ]
    },

});
```

my-app/src/components/Accordian.vue

```
<template>
    <div>
        <p>{{item.title + "Kevin" }}</p>
        <span @click="toggle =! toggle">Details</span>
        <p v-if="toggle">{{item.description}}</p>
        <br>
    </div>
</template>

<script>
export default {
    name: "Accordion",

    props:['item'],

    data(){
        return {
            toggle: false,
        };
    },
}
</script>
```

###