Webpack Introduction

Webpack with Zero-Configuration

1. Create a project name “webpack-introduction”
2. Create a Package.json file

   ```
   npm init
   ```
3. Install webpack, webpack-cli, and webpack-dev-server . you need to install with --save-dev, so these packages get installed as developer dependencies.

   ```
   npm install --save-dev webpack, webpack-cli, and webpack-dev-server
   ```
4. In package.json file in script section
   a. Remove test statement
   b. Add
      i. `"build": "webpack",`
      ii. `"start": "webpack-dev-server --mode development"`
5. In terminal

   ```
   npm start
   ```

   You will see a message in terminal , indicating a address of local server with port no. Click to that address, your app should be running.

   Webpack-dev-server will run the project with live reload and if you made any change in your js file (index.js or greeting.js) that will reflect in browser immediately.
   Webpack-dev-server will also said the js files have been bundled in main.js. But you may not be able to see main.js file in your project (unless you build it first). main.js file will be created on the run and sit on browser memory. You can see it by visiting localhost:/8080/webpack-web-dev/main.js

Install jQuery

To install loader like jquery, on terminal run

```
npm install jquery
```

Note that jquery would installed as dependencies not dev-dependencies

In index.js add

```
let $ = require("jquery");
```

Now you can use jquery as normal in your js code

Install loader

1. To install loader like css , on terminal run

```
npm install --save-dev style-loader css-loader
```

2. Put some style rules on src/css/styles.css
3. Add line

```
require("!style-loader!css-loader!../css/styles.css");
```

4. As first line in index.js file
5. Test on browser, that should work.

But we dont want to do it everytime there is new dependency. For this we need to set a configuration file.

Creating a configuration file

1.Create a file webpack.config.js in project folder ans add

```
module.exports = {
 entry: "./src/js/index.js",
 output: {
   path: __dirname,
   filename: "main.js"
 },
```

entry specify the entry point for your js code and *output* specify your bundled code

2. Now give some rules for css

3. Add in webpack.config.js file

```
module: {
   rules: [
     { test: /\.css$/, loader: "style-loader!css-loader" }
       ]
 }
};
```

`/\.css$/` tells webpack to look for all the file with .css extention

4. Now in the index.js file remove line

```
require("!style-loader!css-loader!../css/styles.css");
```

5. And add

```
require("../css/styles.css");
```

Install Babel

To install babel loader, run in terminal

npm install --save-dev --babel-core --babel-loader babel-preset-es2015

Add another test in `webpack.config.js file after css-test`

```
module: {
   rules: [
     { test: /\.css$/, loader: "style-loader!css-loader" },
     {
       test: /\.js$/,
       loader: "babel-loader",
       exclude: /node_modules/
     }
   ]
 }
```

`/\.js$/` tells webpack to look for all file with .js extension and `exclude: /node_modules/`

Tells to exclude node_modules folder

Here is a little boilerplate with css and jquery and babel