**Basic information (Webpack Tutorial - Replace Gulp or Grunt plugins with a single tool):**

Webpack is a module loader. There is three module loader out there, namely “require.js”, “Browserify” and “Webpack”. “Browserify” and “Webpack” are more popular.

**A simple example:**

* Enter “npm init” to create a package.json file
* Enter “npm install -S webpack”. (This is not installed globally. “-s” represents a different thing.)
* Side note: In npm 1.0, there are two ways to install things:
  + globally—the modules will be dropped in {prefix}/lib/node\_modules and puts executable files in {prefix}/bin, where {prefix} is usually something like /usr/local (on Windows, it is “%AppData%\npm”). It also installs man (manual) pages in {prefix}/share/man, if they’re supplied (on windows, man pages are not installed).
  + locally—this installs your package in the current working directory. Node modules go in “./node\_modules”, executables go in “./node\_modules/.bin/”, and man pages aren’t installed at all.
* Enter “npm install -g webpack” as well.
* Enter “touch webpack.config.js”. “webpack.config.js” will tell the webpack where to look and how to act.

<!-- ./index.html -->

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Some Page</title>

</head>

</html>

<body>

<h1>My Webpack Page</h1>

<script src="js/scripts.min.js"></script>

</body>

/\* ./js/script.js \*/

require(‘./module1.js’);

require(‘./module2.js’);

/\* ./js/module1.js \*/

console.log(“module1 stuff”);

/\* ./js/module2.js \*/

console.log(“module2 stuff”);

/\* ./webpack.config.js (This file can get very complicated) \*/

/\* Is the Node environment “production” (not “debug”)? If so, we will run all the minification things, and we will not do source mapping. Source mapping help the console to log things \*/

var debug = process.env.NODE\_ENV !== "production";

var webpack = require('webpack');

module.exports = {

context: \_\_dirname, // The directory the app “lives” in

devtool: debug ? "inline-sourcemap" : null,

// This is where we start off

// Original version: “entry: "./js/scripts.js"”

entry: \_\_dirname + "/js/scripts.js",

output: {

path: \_\_dirname + "/js",

filename: "scripts.min.js" // We are going to create this file

},

// Here is where Webpack comes in

plugins: debug ? [] : [

// We are going to strip out any duplicate code

new webpack.optimize.DedupePlugin(),

new webpack.optimize.OccurenceOrderPlugin(),

/\* This will help to get rid of source map, comments, etc. so that it is

production ready \*/

new webpack.optimize.UglifyJsPlugin({ mangle: false, sourcemap: false }),

],

};

* Enter “webpack --mode=development” (if it is “webpack”, the mode will be “production”) in the console to build “./js/script.js”, “./js/module1.js” and “./js/module2.js” and emit “./js/scripts.min.js”
* Side note: the following feedback appeared:

One CLI (command line interface) for webpack must be installed. These are recommended choices, delivered as separate packages:

- webpack-cli (https://github.com/webpack/webpack-cli)

The original webpack full-featured CLI.

- webpack-command (https://github.com/webpack-contrib/webpack-command)

A lightweight, opinionated webpack CLI.

We will use "npm" to install the CLI via "npm install -D".

Which one do you like to install (webpack-cli/webpack-command):

(I chose “webpack-cli”)

* Side note: I install webpack-cli globally as well and changed the property “entry” in “webpack.config.js”.
* Check out “scripts.min.js” generated in the location “./js”
  + Some extra code is generated at the front
* Enter “index.html” and check out the Console tab in DevTools (Chrome), two lines are printed out

**The example + jQuery + Lodash:**

* Enter “npm install -S jquery” and “npm install -S lodash”

/\* ./js/module1.js \*/

var $ = require(‘jquery’);

$(‘hi’).html(“new text”);

// Result: the web page shows “new text” instead of “My Webpack Page”

* Run Webpack again
* Enter “$.fn.jquery” in the browser’s console. It is shown that jQuery does not exist in the global scope of the page
* (If another module needed to require jQuery, the code does not get duplicated.)

/\* ./js/module2.js \*/

var \_ = require(‘lodash’); // Lodash is basically underscore.js

// Go to mockaroo.com to get some random data (format: JSON (array))

var people = /\* Paste the array here \*/

// How many female people are there?

var femaleCount = \_.filter(people, {gender: “Female”}).length;

alert(femaleCount + “ females!”); // A dialog box shows up

* Run Webpack again
* jQuery can do “form validation” as well