WEBPACK USER GUIDE

softcontext@gmail.com

2017

https://dinosaurscode.xyz/tutorials/2016/07/24/beginner-friendly-webpack-tutorial/

WEBPACK

Webpack defines itself as a “module bundler.”

What the heck does that mean?

It means less HTTP Get requests in a nutshell.

How?

Node.js, on the backend, allows us to seperate our code into modules to allow easy code reuse and prevent name clashes using CommonJS.

Example:

```
// module1.js
module.exports = {name: "module1"};

// module2.js
module.exports = {name: "module2"};

// app.js
var mod1 = require('module1.js');
var mod2 = require('module2.js');

console.log(mod1.name);
console.log(mod2.name);

#output
mod1
mod2
```

Unfortunately, CommonJS isn't supported on the frontend and thus we're stuck with making several HTTP requests for each piece of code we require.

```
<script src="one.js"></script>
<script src="two.js"></script>
```

And even then we're not guaranteed the order that they'll load in.

Webpack to the rescue

Webpack takes care of this by bundling all your modules into a single .js file.

Getting started

Create a new package.json using npm:

```
$ npm init -y
```

Install webpack locally:

```
$ npm install --save-dev webpack
```

Create a file named bundleme.js:

```
var mod1 = require('./module1');

document.write(mod1.name);
```

module1.js

```
module.exports = {name: "module1"};
```

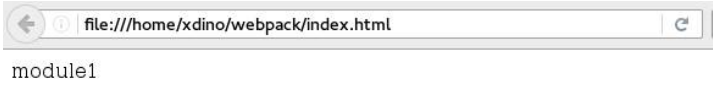
index.html

Lastly create an index.html file to load the bundle in the browser:

```
<!doctype html>
<html>
<body>
  <script src="bundle.js"></script>
</body>
</html>
```

Build it and then open index.html in a browser:

```
$ ./node_modules/.bin/webpack ./bundleme.js bundle.js
```



module1

Note that we installed webpack locally which comes with a prebuilt webpack binary inside the long path above.

Webpack config

Running webpack through the terminal isn't recommended but instead we can write a config file that configures its options.

Create the file webpack.config.js:

```
module.exports = {
    entry: "bundleme.js",
    output: {
      path: __dirname,
      filename: "bundle.js"
    }
}
```

Add an npm script to package.json that runs webpack:

```
"scripts": {
    "build": "webpack"
}
```

Now you can run the npm script and your bundle will build:

```
$ npm run build
```

Loaders

Not only can webpack bundle javascript but it can also do .css with loaders.

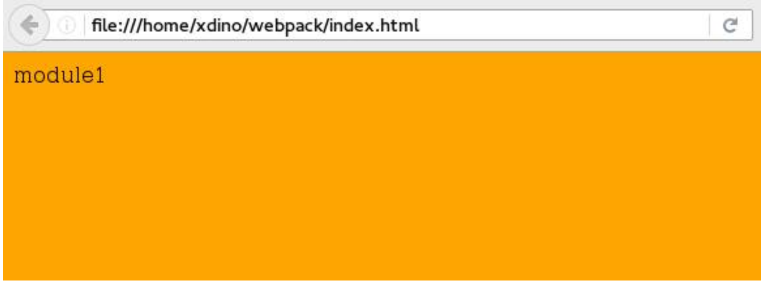
Install css-loader and style-loader:

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

Require a style.css file and pipe it into the loaders:

```
//bundleme.js
require('style!css!./style.css');
var mod1 = require('./module1');

document.write(mod1.name);
```



file:///home/xdino/webpack/index.html

module1

Plugins

Webpack comes with helpful plugins such as compressing javascript.

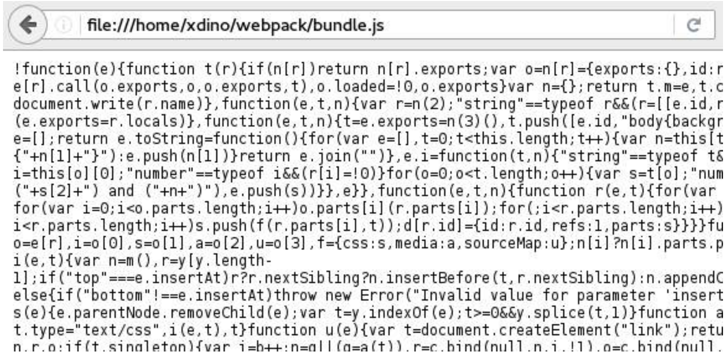
Open webpack.config.js:

```
var webpack = require('webpack');

module.exports = {
    entry: "bundleme.js",
    output: {
      path: __dirname,
      filename: "bundle.js"
    },
    plugins: [
      new webpack.optimize.UglifyJsPlugin()
    ]
}
```

Run the npm build script and your bundle.js is now minified:

```
$ npm run build
```



file:///home/xdino/webpack/bundle.js

```
!function(e){function t(r){if(n[r])return n[r].exports;var o=n[r]={exports:{},id:r
e[r].call(o.exports,o,o.exports,t),o.loaded=!0,o.exports}var n={};return t.m=e,t.c
document.write(r.name)},function(e,t,n){var r=n(2);"string"==typeof r&&(r=[[e.id,r
(e.exports=r.locals)},function(e,t,n){t=e.exports=n(3)(),t.push([e.id,"body{backgr
e=[];return e.toString=function(){for(var e=[],t=0;t<this.length;t++){var n=this[t
{"+n[1]+"}"):e.push(n[1])}return e.join("")},e.i=function(t,n){"string"==typeof t&
i=this[o][0];"number"==typeof i&&(r[i]=!0)}for(o=0;o<t.length;o++){var s=t[o];"num
("+s[2]+") and ("+n+")"),e.push(s))}},e}},function(e,t,n){function r(e,t){for(var
for(var i=0;i<o.parts.length;i++)o.parts[i](r.parts[i]);for(;i<r.parts.length;i++)
i<r.parts.length;i++)s.push(f(r.parts[i],t));d[r.id]={id:r.id,refs:1,parts:s}}}}fu
o=e[r],i=o[0],s=o[1],a=o[2],u=o[3],f={css:s,media:a,sourceMap:u};n[i]?n[i].parts.p
i(e,t){var n=m(),r=y[y.length-
1];if("top"===e.insertAt)r?r.nextSibling?n.insertBefore(t,r.nextSibling):n.appendC
else{if("bottom"!==e.insertAt)throw new Error("Invalid value for parameter 'insert
s(e){e.parentNode.removeChild(e);var t=y.indexOf(e);t>=0&&y.splice(t,1)}function a
t.type="text/css",i(e,t),t}function u(e){var t=document.createElement("link");retu
n,r,o;if(t.singleton){var i=b++;n=g||(g=a(t)),r=c.bind(null,n,i,!1),o=c.bind(null,
```

And that’s the basics of Webpack. From here I suggest heading over to the official docs.

https://webpack.js.org/guides/get-started/

Thanks for reading!