Once installed, you can interface with webpack either from its CLI or API. If you're still new to webpack, please read through the core concepts and this comparison to learn why you might use it over the other tools that are out in the community.

Webpack is used to compile JavaScript modules.

Basic Setup œ

First let's create a directory, initialize npm, install

webpack locally, and install the webpack-cli (the

tool used to run webpack on the command line):

npm install webpack webpack-cli --save-dev

blocks to show you what changes we're making

Now we'll create the following directory structure,

let element = document.createElement('div')

element.innerHTML = _.join(['Hello', 'webpace

document.body.appendChild(component());

<title>Getting Started</title>

<script src="https://unpkg.com/lodash@4.16</pre>

<script src="./src/index.js"></script>

We also need to adjust our package.json file in

private, as well as removing the main entry.

This is to prevent an accidental publish of your

If you want to learn more about the inner

recommend reading the npm documentation.

"test": "echo \"Error: no test specified

workings of package.json, then we

"name": "webpack-demo"

"version": "1.0.0", "description": "", "private": true,

"main": "index.js",

"scripts": {

"keywords": [], "author": "",

"license": "ISC",

"devDependencies": { "webpack": "^4.20.2",

"dependencies": {}

"webpack-cli": "^3.1.2"

In this example, there are implicit dependencies

between the <script> tags. Our index.js file

depends on lodash being included in the page

before it runs. This is because index.js never

explicitly declared a need for lodash; it just

There are problems with managing JavaScript

It is not immediately apparent that the script

If a dependency is missing, or included in the

wrong order, the application will not function

If a dependency is included but not used, the

Let's use webpack to manage these scripts instead.

First we'll tweak our directory structure slightly,

separating the "source" code (/src) from our

"distribution" code (/dist). The "source" code is

"distribution" code is the minimized and optimized

output of our build process that will eventually

9

browser will be forced to download

assumes that the global variable __exists.

depends on an external library.

},

projects this way:

properly.

unnecessary code.

Creating a Bundle

the code that we'll write and edit. The

be loaded in the browser:

webpack-demo

+ |- /dist

|- /src

|- package.json

+ |- index.html |- index.html

|- index.js

npm install --save lodash

To bundle the lodash dependency with

index.js, we'll need to install the library locally:

When installing a package that will be bundled

npm install --save . If you're installing a

linter, testing libraries, etc.) then you should use

let element = document.createElement('div

// Lodash, currently included via a scrip element.innerHTML = _.join(['Hello', 'webp

document.body.appendChild(component());

Now, since we'll be bundling our scripts, we have to

update our index.html file. Let's remove the

lodash <script> ,as we now import it, and

<title>Getting Started</title>

<script src="main.js"></script>

In this setup, index. is explicitly requires

global scope pollution). By stating what

lodash to be present, and binds it as _ (no

dependencies a module needs, webpack can use

this information to build a dependency graph. It

bundle where scripts will be executed in the correct

With that said, let's run npx webpack, which will

take our script at src/index.js as the entry

point, and will generate dist/main.js as the

Node 8.2/npm 5.2.0 or higher, runs the webpack

webpack package we installed in the beginning:

Built at: 13/06/2018 11:52:07

Asset Size Chunks

WARNING in configuration

main.js 70.4 KiB

text: 'Hello webpack'.

Modules

API for more information.

loader system.

project

binary(./node_modules/.bin/webpack) of the

The 'mode' option has not been set, webpack w You can also set it to 'none' to disable any

Your output may vary a bit, but if the build is

Open index.html in your browser and, if

90

webpack supports modules out of the box.

successful then you are good to go. Also, don't

worry about the warning, we'll tackle that later.

everything went right, you should see the following

The import and export statements have been

standardized in ES2015 and are supported in most

browsers. Some older browsers still lag behind but

Behind the scenes, webpack actually "transpiles"

the code so that older browsers can also run it. If

see how webpack does this, it's quite ingenious!

you inspect dist/main.js, you might be able to

Besides import and export, webpack supports

various other module syntaxes as well, see Module

Note that webpack will not alter any code other

using other ES2015 features, make sure to use a

transpiler such as Babel or Bublé via webpack's

Using a Configuration

As of version 4, webpack doesn't require any

configuration, but most projects will need a more

complex setup, which is why webpack supports a

configuration file. This is much more efficient than

having to manually type in a lot of commands in

the terminal, so let's create one:

webpack-demo

|- /dist

- /src

webpack.config.js

output: {

new configuration file:

main.js 70.4 KiB

WARNING in configuration

Asset

}

};

|- package.json

+ |- webpack.config.js

|- index.html

|- index.js

module.exports = {

const path = require('path');

entry: './src/index.js',

filename: 'main.js',

path: path.resolve(__dirname, 'dist')

Now, let's run the build again but instead using our

The 'mode' option has not been set, webpack w You can also set it to 'none' to disable any

If a webpack.config.js is present, the

webpack command picks it up by default. We

use the --config option here only to show

will be useful for more complex configurations

A configuration file allows far more flexibility than

simple CLI usage. We can specify loader rules,

enhancements this way. See the configuration

90

Given it's not particularly fun to run a local copy of

shortcut. Let's adjust our package.json by adding

"test": "echo \"Error: no test specified

webpack from the CLI, we can set up a little

"name": "webpack-demo",

"build": "webpack"

"version": "1.0.0", "description": "",

"scripts": {

"keywords": [],

"license": "ISC",

"dependencies": {

"devDependencies": {

"webpack": "^4.20.2",

"lodash": "^4.17.5"

"webpack-cli": "^3.1.2"

Now the npm run build command can be used

locally installed npm packages by name the same

standard in most npm-based projects because it

common scripts (each with flags like --config if

Now run the following command and see if your

Size Chunks

The 'mode' option has not been set, webpack w You can also set it to 'none' to disable any

Custom parameters can be passed to webpack

by adding two dashes between the npm run

build command and your parameters, e.g.

90

Now that you have a basic build together you

Management to learn how to manage assets like

images and fonts with webpack. At this point, your

If you're using npm 5, you'll probably also see a

package-lock.json file in your directory.

If you want to learn more about webpack's design,

configuration pages. Furthermore, the API section

EugeneHlushko

aaronang

bebraw

chrisVillanueva

jecoopr

simon04

varunjayaraman

COMPARISON

GET STARTED ORGANIZATION STARTER KITS

GITTER

CHANGELOG

BRANDING

TheDutch(

digs into the various interfaces webpack offers.

you can check out the basic concepts and

should move on to the next guide Asset

npm run build -- --colors .

Chunk Na

0 [emitted] main

in place of the npx command we used earlier.

Note that within scripts we can reference

way we did with <code>npx</code> . This convention is the

allows all contributors to use the same set of

"author": "",

},

necessary).

script alias works:

npm run build

main.js 70.4 KiB

Conclusion

project should look like this:

project

webpack-demo

|- /dist

- /src

|- package.json

|- main.js

|- index.js |- /node_modules

Contributors

Vanguard90

bjarki

cntanglijun

johnstew

sudarsangp

ztomasze

GLOSSARY

|- index.html

|- webpack.config.js

WARNING in configuration

Asset

plugins, resolve options and many other

documentation to learn more.

NPM Scripts

an npm script:

package.json

that you can pass a config of any name. This

that need to be split into multiple files.

npx webpack --config webpack.config.js

Size Chunks

than import and export statements. If you are

90

Chunk Na

main

0 [emitted]

Chunk Na

main

[emitted]

output. The npx command, which ships with

then uses the graph to generate an optimized

<script src="https://unpkg.com/lodash@4.</pre>

<script src="./src/index.js"></script>

modify the other <script> tag to load the

bundle, instead of the raw /src file:

dist/index.html

<html>

<head>

</head> <body>

</body>

</html>

order.

npx webpack

<!doctype html>

into your production bundle, you should use

package for development purposes (e.g. a

npm install --save-dev . More

information can be found in the npm

Now, lets import lodash in our script:

+ import _ from 'lodash';

function component() {

return element;

documentation.

src/index.js

project

order to make sure we mark our package as

mkdir webpack-demo && cd webpack-demo

Throughout the Guides we will use diff

to directories, files, and code.

files and their contents:

webpack-demo

|- index.html

+ |- /src

src/index.js

}

index.html

<html>

</head>

</body>

</html>

code.

package.json

<!doctype html>

- package.json

|- index.js

function component() {

return element;

project

npm init -y