



The Next Generation ⚡ of Web Bundling

with Rspack

Konstantinos Leimonis

Software Engineer



What's a bundler?

browserify

code
articles
find modules

require('modules') in the browser

Write your browser code with `node.js`-style requires:

main.js

```
var foo = require('./foo');  
var gamma = require('gamma');  
  
var n = gamma(foo(5) * 3);  
var txt = document.createTextNode(n);  
document.body.appendChild(txt);
```

Export functionality by assigning onto `module.exports` or `exports`:

foo.js

```
module.exports = function (n) { return n * 11 }
```

Install modules with `npm`:

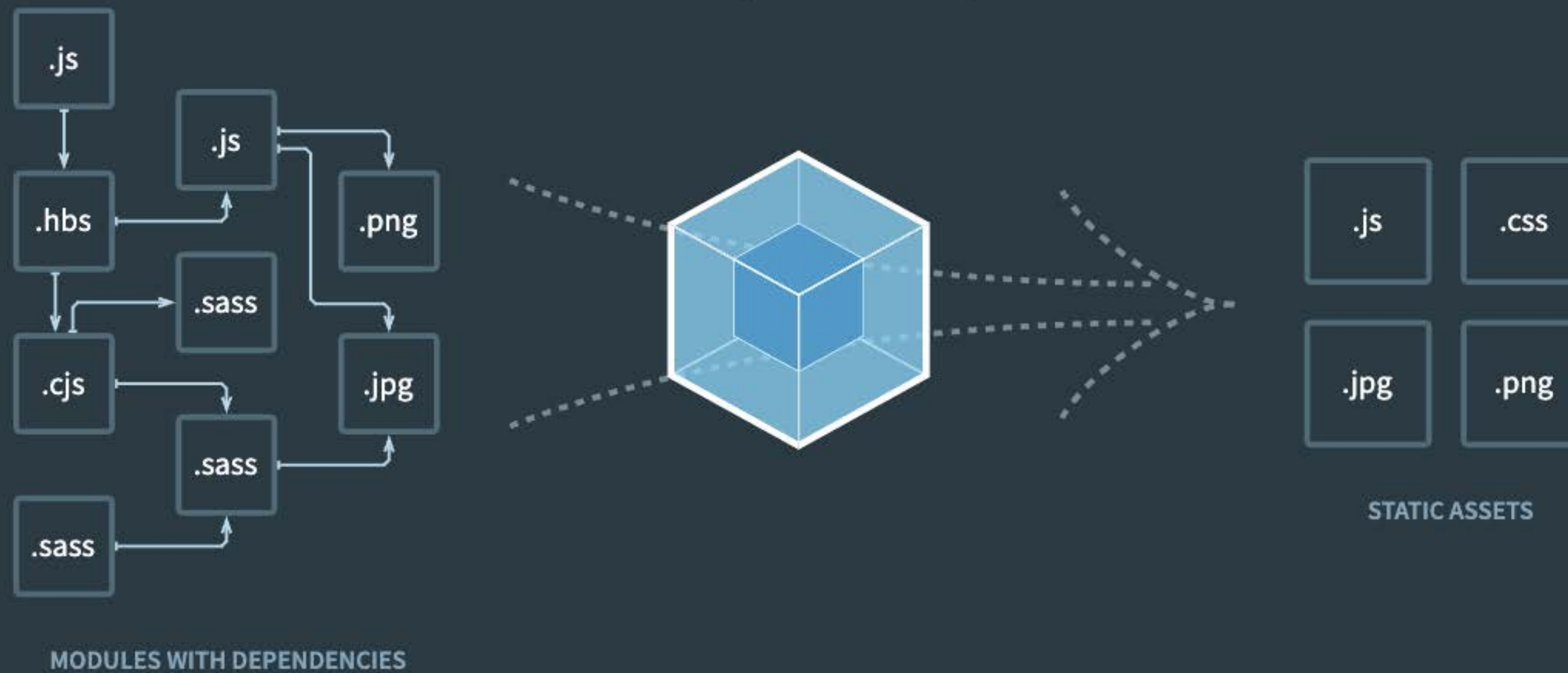
```
npm install gamma
```

```
var MyModule = 'hello world';

if (typeof module !== 'undefined' && module.exports) {
  module.exports = MyModule;
} else {
  (typeof self !== 'undefined' ? self : window).MyModule = MyModule;
}
```

Webpack

bundle your scripts



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

```
var svg = require("svg-url?limit=1024!./file.svg");
```

webpack v5.94.0

Packs ECMAScript/CommonJs/AMD modules for the browser. Allows you to split your codebase into multiple bundles, which can be loaded on demand. Supports loaders to preprocess files, i.e. json, jsx, es7, css, less, ... and your custom stuff.

github.com/webpack/webpack

5,594,158,987 total npm downloads

month

week



6000 packages found

1 2 3 ... 300 »

Sort Packages

Optimal

Popularity

Quality

Maintenance

webpack exact match

Packs ECMAScript/CommonJs/AMD modules for the browser demand. Supports loaders to preprocess files, i.e. json, js

evilebottnawi published 5.94.0 • a month ago

to split your codebase into multiple bundles, which can be loaded on ... and your custom stuff.



sass-loader

Sass loader for webpack

sass libsass webpack loader

evilebottnawi published 16.0.2 • 3 days ago

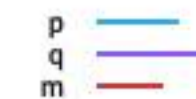


webpack-dev-middleware

A development middleware for webpack

webpack middleware development

evilebottnawi published 7.4.2 • a month ago



Contentful
https://www.contentful.com › blog › putting-your-web...

How to put your Webpack bundle on a diet

7 Dec 2022 — The bundle now takes up an enormous 744.4 KiB. This is because Webpack imports every Moment locale, regardless of whether it's used, by default.

GitHub
https://github.com › webpack › webpack › issues

[webpack 5] rebuild is twice as slow comparing to webpack 4

2 Dec 2020 — After upgrading to webpack 5 in our project we noticed that rebuild time increased more than 2 times comparing to webpack 4.

DEV Community
https://dev.to › slashgear_ › how-to-boost-the-speed-of-...

How to boost the speed of your webpack build?

4 Mar 2020 — Use Speed Measure Plugin to debug webpack build time · Track your build time evolution to detect big evolution before merge · Follow webpack ...

Stack Overflow
https://stackoverflow.com › questions › webpack-taking...

Webpack: Taking a long time to build

As discussed in the comments, here are the most obvious changes you could make to speed up your build: UglifyJsPlugin and ExtractTextPlugin ...

webpack's compile time is very slow

Asked 6 years, 3 months ago Modified 4 months ago Viewed 29k times

My webpack is going very slow when it starts and when theres a change - compiling. Actually making development very slow right now. I'm only using greensock as the vendor but nothing else. I'm using only a few images too.. not sure.

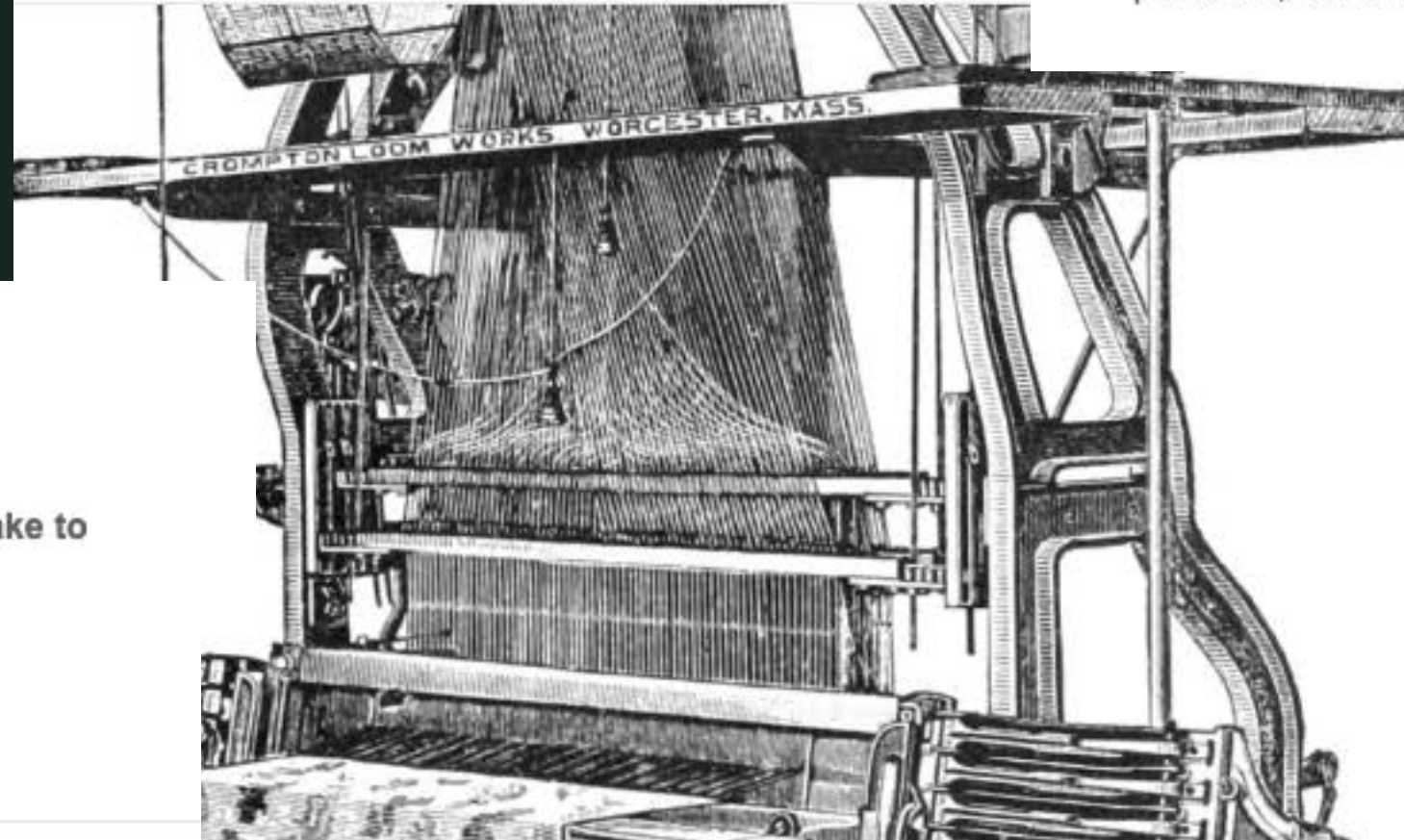
Here is the code:

```
const path = require('path');
var webpack = require('webpack');
var HtmlWebpackPlugin = require('html-webpack-plugin');
var ExtractTextPlugin = require('extract-text-webpack-plugin');

// const ASSET_PATH = process.env.ASSET_PATH || '/';
// publicPath: '/dist'

var isProd = process.env.NODE_ENV === 'production';
var cssDev = ['style-loader', 'css-loader', 'sass-loader'];
```

slack engineering



The build tools of yore: a Loom with Jacquard machine attached

useanvil.com
https://www.useanvil.com › blog › engineering › mini...

Minimizing Webpack bundle size - Anvil

26 May 2021 — Learn how to minimize your Webpack bundle size by following these best practices, ensuring an optimal user experience.

Keep webpack Fast: A Field Guide for Better Build Performance

Rowan Oulton Staff Engineer - Front End

13 minutes • Written 6 years ago

Twitter Facebook LinkedIn

Search

webpack is a brilliant tool for bundling frontend assets. When things start to slow down, though, its batteries-included nature and the ocean of third-party tooling can make it difficult to optimize. Poor performance is the norm and not the exception. But it doesn't have to be that way, and so—after many hours of research, trial, and error—what follows is a field guide offering up what we learned on our path towards a faster build.

Post Types

Post (171)

Categories

Uncategorized (149)



/'arespæk/



 High performance

 Fast Startup time

 Lightning fast HMR

 Webpack Compatibility

 Framework Agnostic





Jakub Romańczyk

@_jbroma





Re.Pack 5 Alpha is here and with it:

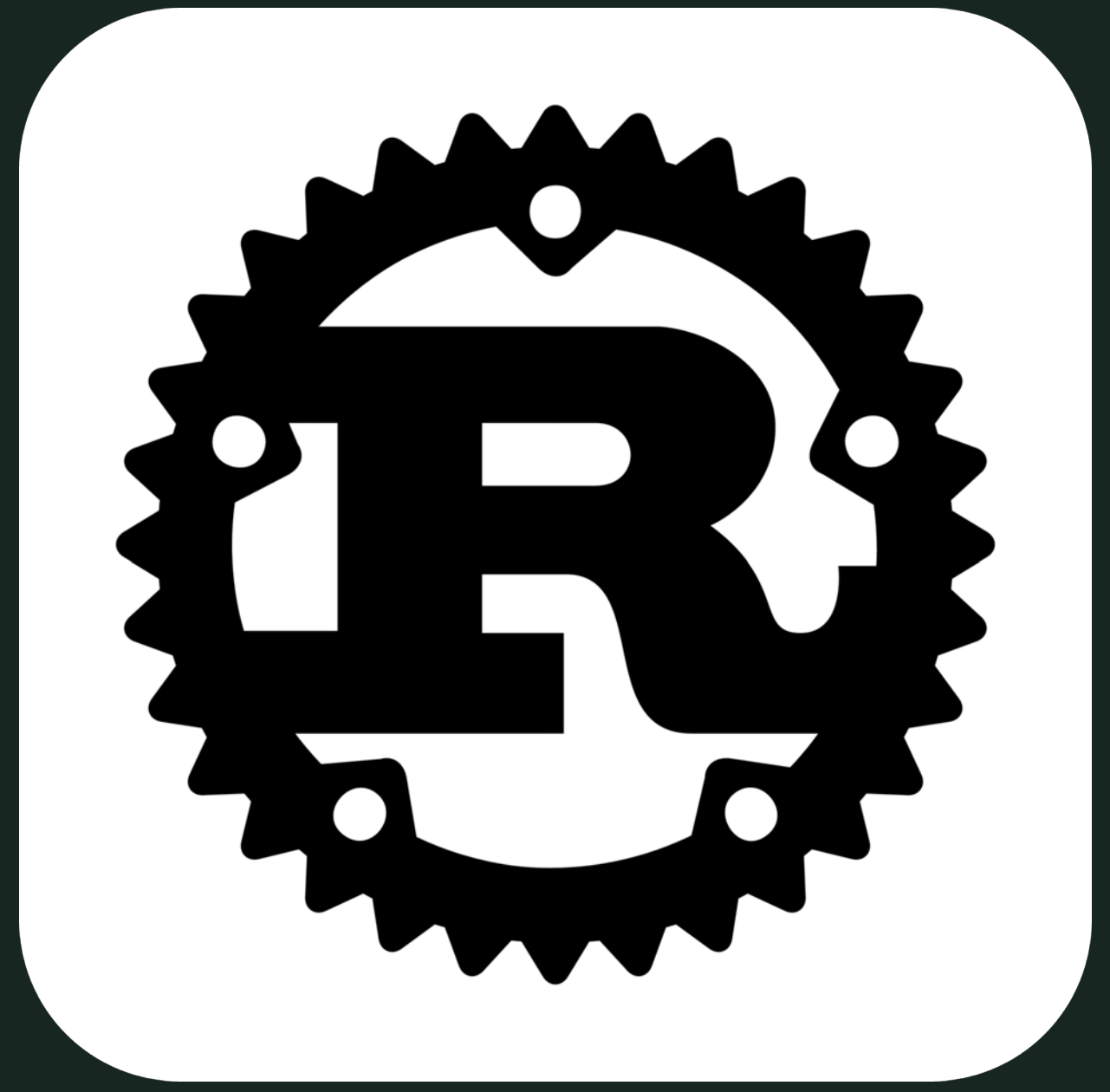
🔥 5x faster builds

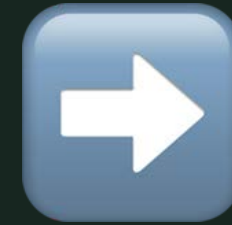
🔥 5x faster HMR

All thanks to [@rspack_dev](#) 🦀 under the hood!
(don't worry, webpack is still supported)

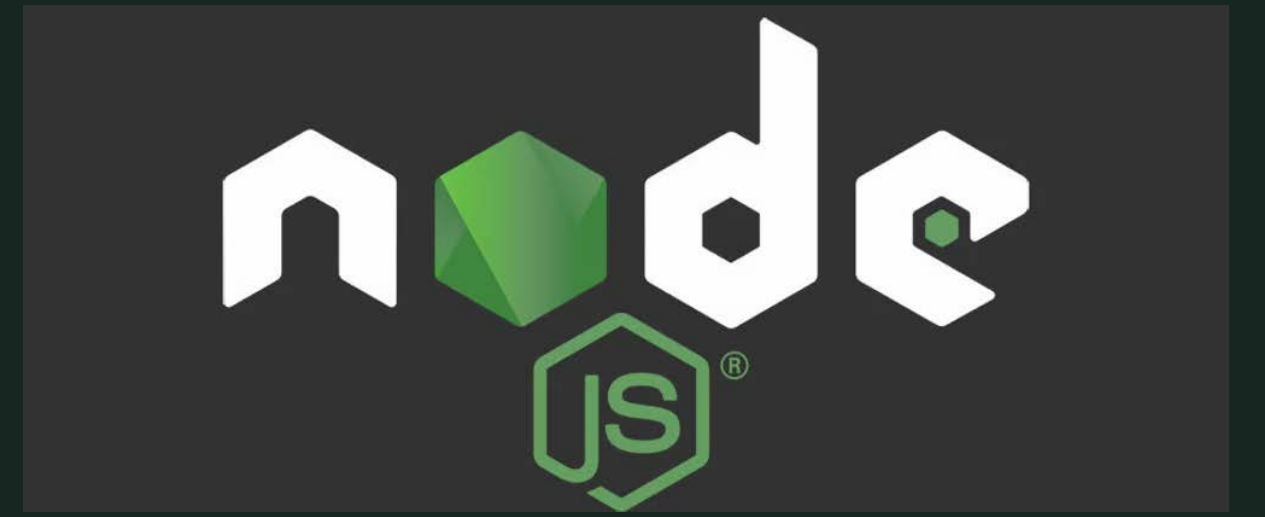
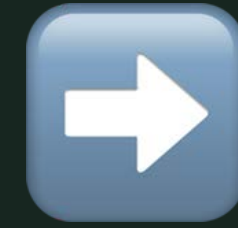
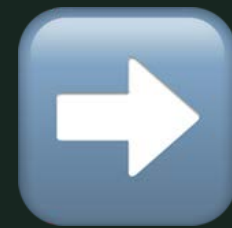
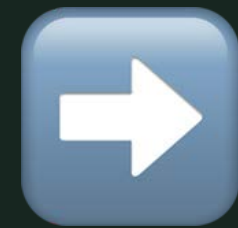
👉 Give it a try today and help us get to the stable release!

RE.PACK + 	RE.PACK + 
<pre>> pnpm react-native bundle</pre>	<pre>> pnpm react-native bundle</pre>
Re.Pack 5.0.0-alpha.0	Re.Pack 5.0.0-alpha.0
<hr/>	<hr/>
📦 Assets: 17	📦 Assets: 17
📄 Modules: 685	📄 Modules: 685
🕒 Time: 7.70s	🕒 Time: 1.75s
<hr/>	<hr/>









Features

Features

- CSS *
- Asset Management
- JSON
- Web workers



Features

- Loaders
- Plugins
- HMR
- Optimisations
 - Code Splitting
 - Tree shaking
- Module Federation



Migration

Dependencies

```
pnpm add @rspack/core @rspack/cli -D
```

```
pnpm remove webpack webpack-cli webpack-dev-server
```



```
// package.json
{
  "scripts": {
    "serve": "webpack serve",
    "build": "webpack build"
  }
}
```

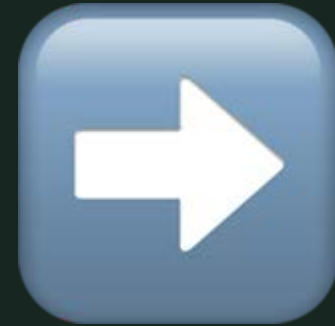
```
// package.json
{
  "scripts": {
    "serve": "rspack serve",
    "build": "rspack build"
  }
}
```

```
mv webpack.config.js rspack.config.js
```



Loaders

BABEL



babel-loader

builtin:swc-loader

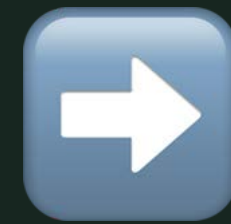
```
// webpack.config.js

module.exports = {
  module: {
    rules: [{
      test: /\.tsx?$/,
      use: [
        {
          loader: 'babel-loader',
          options: {
            presets: ['@babel/preset-typescript'],
          },
        },
      ],
    },
    {
      test: /\.jsx?$/,
      use: [
        {
          loader: 'babel-loader',
          options: {
            presets: ['@babel/preset-react'],
          },
        },
      ],
    },
  ],
}
```

```
// rspack.config.js

module.exports = {
  module: {
    rules: [{
      test: /\.jsx?(tsx?)$/,
      exclude: [/[\w/]node_modules[\w/] /],
      loader: "builtin:swc-loader",
      options: {
        sourceMap: true,
        jsc: {
          parser: {
            syntax: "typescript",
            tsx: true,
          },
          transform: {
            react: {
              runtime: "automatic",
              development: isDevelopment,
            },
          },
        },
      },
    },
    {
      env: {
        targets: ["browserlist"],
      },
    },
  ],
},
};
```


file-loader



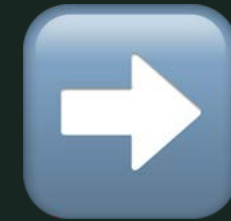
asset / resource

url-loader



asset / inline

raw-loader



asset / source

Plugins

```
// webpack.config.js

const webpack = require('webpack');

module.exports = {
  // ...
  plugins: [
    new webpack.DefinePlugin({
      // ...
    }),
  ],
}
```

```
// rspack.config.js

const rspack = require('@rspack/core');

module.exports = {
  // ...
  plugins: [
    new rspack.DefinePlugin({
      // ...
    }),
  ],
}
```

```
// rspack.config.js

const rspack = require('@rspack/core');
const HtmlWebpackPlugin = require("html-webpack-plugin");

module.exports = {
  // ...
  plugins: [
    new HtmlWebpackPlugin({
      // ...
    }),
  ],
}
```

```
// rspack.config.js

const rspack = require('@rspack/core');

module.exports = {
  // ...
  plugins: [
    new rspack.HtmlRspackPlugin({
      // ...
    }),
  ],
}
```

<https://rspack.dev/guide/compatibility/plugin>

The screenshot shows the Rspack documentation website. The navigation bar includes 'Rspack', a search bar, and links for 'Guide', 'Config', 'Plugin' (highlighted), 'API', 'Blog', 'About', 'Contribute', 'Version', and social media icons. The left sidebar lists 'Webpack-aligned built-in plugins' with 'Overview' selected. The main content area is titled 'Overview' and contains a paragraph explaining the table's purpose. The table below lists 18 plugins, all of which are supported.

Plugin	Support status	Notes
BannerPlugin	Supported	
EnvironmentPlugin	Supported	
HotModuleReplacementPlugin	Supported	
IgnorePlugin	Supported	
LimitChunkCountPlugin	Supported	
ModuleFederationPlugin	Supported	
NormalModuleReplacementPlugin	Supported	
ProvidePlugin	Supported	
SourceMapDevToolPlugin	Supported	
NodeTemplatePlugin	Supported	
NodeTargetPlugin	Supported	
ElectronTargetPlugin	Supported	
EnableChunkLoadingPlugin	Supported	
EnableLibraryPlugin	Supported	
DefinePlugin		
ProvidePlugin		
BannerPlugin		
HotModuleReplacementPlugin		
IgnorePlugin		
ProgressPlugin		
ExternalsPlugin		
EvalSourceMapDevToolPlugin		
SourceMapDevToolPlugin		
SplitChunksPlugin		
NodeTargetPlugin		
NodeTemplatePlugin		
EnableChunkLoadingPlugin		
EnableLibraryPlugin		
EnableWasmLoadingPlugin		
ModuleFederationPlugin		
ModuleFederationPluginV1		
EnvironmentPlugin		
LimitChunkCountPlugin		
NormalModuleReplacementPlugin		
JavascriptModulesPlugin		

Caveats

(S)CSS Modules

WARNING in ./src/App.tsx

⚠ HarmonyLinkingWarning: export 'default' (imported as 'styles') was not found in './App.module.scss'
k, logo, logo-spin)

```
[38:31]
36 |         }, this),
37 |         /**#__PURE__*/ _jsxDEV("a", {
38 |           className: styles.link,
39 |           href: "https://reactjs.org",
40 |           target: "_blank",
```

```
import styles from './App.module.scss'
```

```
const App = () => {  
  return (  
    <div className={styles.container}>  
      <header className={styles.header}>  
        ...  
      </header>  
    </div>  
  )  
}
```

```
import * as styles from './App.module.scss'
```

```
const App = () => {  
  return (  
    <div className={styles.container}>  
      <header className={styles.header}>  
        ...  
      </header>  
    </div>  
  )  
}
```

```
// rspack.config.js
```

```
module.exports = {  
  ...  
  experiments: {  
    css: true,  
  },  
}
```

```
// rspack.config.js

module.exports = {
  ...
  module: {
    generator: {
      'css/auto': {
        esModule: false,
        exportsConvention: 'camel-case-only',
        localIdentName: '[path][name][ext]__[local]',
      },
      css: {
        esModule: false,
        exportsConvention: 'camel-case-only',
        localIdentName: '[path][name][ext]__[local]',
      },
      'css/module': {
        esModule: false,
        exportsConvention: 'camel-case-only',
        localIdentName: '[path][name][ext]__[local]',
      },
    },
  },
  parser: {
    'css/auto': {
      namedExports: false,
    },
    css: {
      namedExports: false,
    },
    'css/module': {
      namedExports: false,
    },
  },
},
}
```

```
// rspack.config.js

module.exports = {
  module: {
    rules: [
      ...
      {
        test: /\.scss$/,
        loader: "sass-loader",
        options: {
          api: "modern-compiler",
        },
        type: "css/auto",
      },
      {
        test: /\.css$/,
        type: "css",
      },
    ],
  },
};
```

```
// rspack.config.js

module.exports = {
  module: {
    rules: [
      ...
      {
        test: /\.scss$/,
        loader: "sass-loader",
        options: {
          api: "modern-compiler",
        },
        type: "css/auto",
      },
    ],
  },
};
```


Open

4 of 11 tasks

improve `experiments.css` support #5340

ahabhgk opened this issue on Jan 16 · 9 comments



leimonio commented 4 days ago · edited

Contributor

When CSS Modules are enabled, the styles are usually imported using the default export of the style module, especially in a React and Vue application context. Instead in the current CSS support the `namedExport` is the default configuration and expected behaviour.

<https://nextjs.org/docs/app/building-your-application/styling/css#css-modules>

<https://vuejs.org/api/sfc-css-features.html#css-modules>

```
import React from 'react'
import styles from './App.module.scss'

const App = () => {
  return (
    <div className={styles.container}>
      <header className={styles.header}>
        ...
      </header>
    </div>
  )
}

export default App
```

In that case I'd very much prefer this `module.parser` to be provided along with `css/scss` modules, a default configuration.

```
module.exports = {
  ...
  experiments: {
```



Konstantinos Leimonis

@leimonio



Just out of a client meeting with [@itaymendel](#). In less than 10 minutes, we migrated their 191 Bit components from Webpack to [@rspack_dev](#), drastically improving build time - at least 3x faster! 🕒

Config Mutator

```

const path = require('path');
const HtmlWebpackPlugin = require("html-webpack-plugin");

const isDevelopment = process.env.NODE_ENV === 'development';

module.exports = {
  entry: './src/index.tsx',
  output: {
    path: path.resolve(__dirname, 'dist'),
    filename: 'bundle.js',
    publicPath: '/',
  },
  module: {
    rules: [
      {
        test: /\.?(tsx?)$/,
        exclude: [/[\w]node_modules[\w]/],
        loader: "builtin:swc-loader",
        options: {
          sourceMap: true,
          jsc: {
            parser: {
              syntax: "typescript",
              tsx: true,
            },
            transform: {
              react: {
                runtime: "automatic",
                development: isDevelopment,
              },
            },
          },
          env: {
            targets: 'Chrome >= 48',
          },
        },
      },
      {
        test: /\.jsx?$/,
        exclude: /node_modules/,
        use: {
          loader: 'babel-loader',
          options: {
            presets: ['@babel/preset-env', '@babel/preset-react'],
          },
        },
      },
      {
        test: /\.scss$/,
        use: [
          'style-loader',
          {
            loader: 'css-loader',
            options: {
              importLoaders: 1,
              esModule: false,
            },
          },
          'sass-loader',
        ],
      },
      {
        test: /\.css$/,
        exclude: /\.module\.css$/,
        use: [
          'style-loader',
          'css-loader',
        ],
      },
    ],
  },

```

```

{
  test: /\.svg$/,
  use: [
    {
      loader: require.resolve('@svgr/webpack'),
      options: {
        prettier: false,
        svgo: false,
        svgoConfig: {
          plugins: [{ removeViewBox: false }],
        },
        titleProp: true,
        ref: true,
        typescript: true,
        ext: "tsx",
      },
    },
    {
      loader: require.resolve('file-loader'),
      options: {
        name: 'static/media/[name].[hash].[ext]',
      },
    },
  ],
  issuer: {
    and: [/\.?(ts|tsx|js|jsx|md|mdx)?$/],
  },
},
],
},
resolve: {
  extensions: ['.ts', '.tsx', '.js'],
},
plugins: [
  new HtmlWebpackPlugin({
    template: "public/index.html",
  })
],
};

```

```
// rspack.config.ts

import { RspackConfigMutator }
  from '@teambit/rspack.modules.rspack-config-mutator';

const mutator = new RspackConfigMutator();

const { config } = mutator
  .setMode('development')
  .addTypescript({ react: true })
  .addCssModules();

export default config;
```

```
// rspack.config.ts

import { RspackConfigMutator }
  from '@teambit/rspack.modules.rspack-config-mutator';

const myConfig = { mode: 'development' };

const mutator = new RspackConfigMutator(myConfig);

const { config } = mutator
  .addTypescript({ react: true })
  .addCssModules();

export default config;
```

<https://bit.cloud/teambit/rspack/modules/rspack-config-mutator>

The screenshot shows the Bit Cloud interface for the Rspack Config Mutator module. The top navigation bar includes the 'rspack' logo, tabs for 'Overview', 'Preview', 'Graph', 'Tests', and 'API Reference', and utility icons for code, refresh, clock, and settings. A 'Use' button and a 'workspace' dropdown are also visible. The main content area features the title 'Rspack Config Mutator' and a subtitle 'Rspack configuration toolbox.' Below this are tags for 'RSPACK', 'CONFIGURATION', 'BUNDLER', and 'RUST'. A search bar with 'Component ID' and 'Package' filters contains the text 'teambit.rspack/modules/rspack-config-mutator'. A description states: 'This utility helps you safely create and change your Rspack configuration.' A 'Features:' section lists: 'Add default typescript configuration', 'Add default CSS Modules configuration', 'Add/Remove plugins', and 'Merge custom configuration Check the API description for more.' A 'Usage' section is followed by the heading 'Create a new Rspack configuration' and a code block showing the following TypeScript code:

```
// rspack.config.ts

import { RspackConfigMutator } from '@teambit/rspack.modules.rspack-config-mutator';

const mutator = new RspackConfigMutator();

const { config } = mutator
  .setMode('development')
  .addTypescript({ react: true})
```

Rspack Toolbox



Rspack

A high performance JavaScript bundler written in Rust, with a webpack-compatible API.



Rsbuild

An Rspack-based build tool that provides out-of-the-box setup for enjoyable development experience.



Rspress

A static site generator based on Rsbuild and MDX for creating elegant documentation sites.



Rsdoctor

A powerful one-stop build analyzer for visualizing the build process and build artifacts.



Rslib

A library build tool powered by Rsbuild for developing libraries or UI components.

That's all Folks!