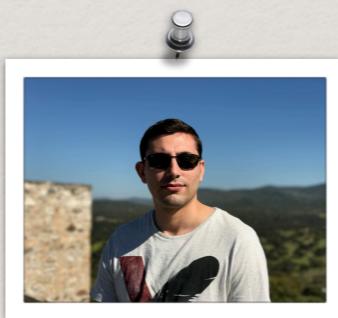




webpack



Alberto Arana Boza



@aaranaboza



albertoarana



svjs-webpack

¿Qué es webpack?

webpack es un paquete de módulos para aplicaciones basadas en Javascript

- ❖ Genera gráficos de dependencias
- ❖ Empaque la los módulos una sola vez en uno o mas ficheros

```
module.exports = {
  devtool: 'cheap-module-source-map',
  entry: [
    require.resolve('./polyfills'),
    require.resolve('react-dev-utils/webpackHotDevClient'),
    paths.appIndexJs,
  ],
  output: {
    path: paths.appBuild,
    pathinfo: true,
    filename: 'static/js/bundle.js',
    chunkFilename: 'static/js/[name].chunk.js',
    publicPath: publicPath,
    devtoolModuleFilenameTemplate: info =>
      path.resolve(info.absoluteResourcePath)
        .replace(/\\/g, '/'),
  },
  resolve: {
    modules: ['node_modules', paths.appNodeModules].concat(
      process.env.NODE_PATH.split(path.delimiter)
        .filter(Boolean)
    ),
    extensions: ['.web.js', '.js', '.json', '.web.jsx', '.jsx'],
    alias: {
      'react-native': 'react-native-web',
    },
    plugins: [
      new ModuleScopePlugin(paths.appSrc, [paths.appPackageJson]),
    ],
  },
},
```

```
module: {
  strictExportPresence: true,
  rules: [
    {
      test: /\.(js|jsx)$/,
      include: paths.appSrc,
      loader: require.resolve('babel-loader'),
      options: {
        cacheDirectory: true,
      },
    },
  ],
},
plugins: [
  new InterpolateHtmlPlugin(env.raw),
  new HtmlWebpackPlugin({
    inject: true,
    template: paths.appHtml,
  }),
  new webpack.NamedModulesPlugin(),
  new webpack.DefinePlugin(env.stringifyed),
  new webpack.HotModuleReplacementPlugin(),
  new CaseSensitivePathsPlugin(),
  new WatchMissingNodeModulesPlugin(paths.appNodeModules),
  new webpack.IgnorePlugin(/\^\.\/locale$/, /moment$/),
],
```

Arquitectura

- ❖ Entornos
 - webpack.config.dev.js
 - webpack.config.prod.js
 - webpack.config.staging.js
 - etc
- ❖ Ficheros Auxiliares
 - env.js
 - paths.js
 - scripts
 - start.js
 - build.js
 - ...
- ❖ package.json

Front-end

Configuración

❖ bail

❖ devtool

❖ entry

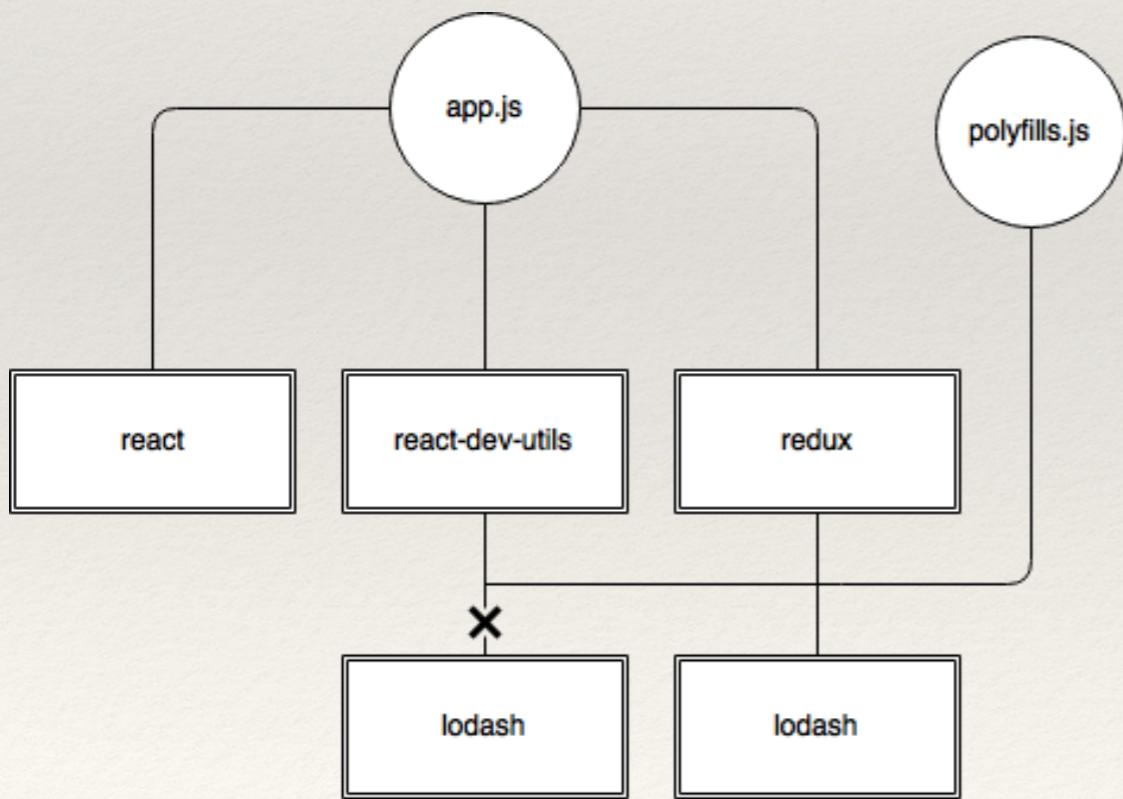
❖ output

```
// This is the production configuration.  
// It compiles slowly and is focused on producing a fast and minimal bundle.  
// The development configuration is different and lives in a separate file.  
module.exports = {  
  // Don't attempt to continue if there are any errors.  
  bail: true,  
  // We generate sourcemaps in production. This is slow but gives good results.  
  // You can exclude the *.map files from the build during deployment.  
  devtool: shouldUseSourceMap ? 'source-map' : false,  
  // In production, we only want to load the polyfills and the app code.  
  entry: [require.resolve('./polyfills'), paths.appIndexJs],  
  output: {  
    // The build folder.  
    path: paths.appBuild,  
    // Generated JS file names (with nested folders).  
    // There will be one main bundle, and one file per asynchronous chunk.  
    // We don't currently advertise code splitting but Webpack supports it.  
    filename: 'static/js/[name].[chunkhash:8].js',  
    chunkFilename: 'static/js/[name].[chunkhash:8].chunk.js',  
    // We inferred the "public path" (such as / or /my-project) from homepage.  
    publicPath: publicPath,  
    // Point sourcemap entries to original disk location (format as URL on Windows)  
    devtoolModuleFilenameTemplate: info => path  
      .relative(paths.appSrc, info.absoluteResourcePath)  
      .replace(/\\"g, '/')  
  },  
};
```

Configuración

❖ entry

```
// In production, we only want to load the polyfills and the app code.  
entry: [  
  require.resolve('./polyfills'),  
  paths.appIndexJs  
,
```



❖ output

```
output: {  
  // The build folder.  
  path: paths.appBuild,  
  // Generated JS file names (with nested folders).  
  // There will be one main bundle, and one file per asynchronous chunk.  
  // We don't currently advertise code splitting but Webpack supports it.  
  filename: 'static/js/[name].[chunkhash:8].js',  
  chunkFilename: 'static/js/[name].[chunkhash:8].chunk.js',  
  // We inferred the "public path" (such as / or /my-project) from homepage.  
  publicPath:  
    // Point sourcemap entries to original disk location (format as URL on Windows)  
    devtoolModuleFilenameTemplate: info => path  
      .relative(paths.appSrc, info.absoluteResourcePath)  
      .replace(/\\/g, '/')  
,
```

▶ publicPath

```
const publicPath = 'localhost/app';  
  
<a href='/contact'> </a>  
  
<a href='localhost/app/contact'> </a>
```

Configuración

- ❖ resolve
- ❖ extensions
- ❖ alias
- ❖ plugins

```
resolve: {  
  // This allows you to set a fallback for where Webpack should look for modules.  
  // We placed these paths second because we want `node_modules` to "win"  
  // if there are any conflicts. This matches Node resolution mechanism.  
  modules: ['node_modules', paths.appNodeModules].concat(  
    // It is guaranteed to exist because we tweak it in `env.js`  
    process.env.NODE_PATH.split(path.delimiter).filter(Boolean)  
,  
  // These are the reasonable defaults supported by the Node ecosystem.  
  // We also include JSX as a common component filename extension to support  
  // some tools, although we do not recommend using it, see:  
  // `web` extension prefixes have been added for better support  
  // for React Native Web.  
  extensions: ['.web.js', '.js', '.json', '.web.jsx', '.jsx'],  
  alias: {  
    'bootstrapp': 'bootstrap/dist/bootstrap',  
  },  
  plugins: [  
    // Prevents users from importing files from outside of src/ (or node_modules/).  
    // This often causes confusion because we only process files within src/ with babel.  
    // To fix this, we prevent you from importing files out of src/ -- if you'd like to,  
    // please link the files into your node_modules/ and let module-resolution kick in.  
    // Make sure your source files are compiled, as they will not be processed in any way.  
    new ModuleScopePlugin(paths.appSrc, [paths.appPackageJson]),  
  ],  
},
```

► resolve

```
/* before */  
import commons from '../..../..//commons'  
  
/* after */  
import commons from 'commons'
```

► alias

```
/* before */  
import bootstrapp from 'bootstrapp/dist/bootstrapp'  
import utils from 'commons/tools/extra/utils'  
  
/* after */  
import bootstrapp from 'bootstrapp'  
import utils from 'utils'
```

Loaders

- ❖ url-loader
 - ❖ babel-loader
 - ❖ file-loader
 - ❖ svg-loader
 - ❖ style-loader
 - ❖ ...
- ❖ css-loader
 - ❖ sass-loader
 - ❖ less-loader
 - ❖ postcss-loader
 - ❖ stylus-loader

```
// "url" loader works just like "file" loader
// but it also embeds
// assets smaller than specified size as data URLs
// to avoid requests.
{
  test: [/\.\bmp$/, /\.\gif$/, /\.\jpe?g$/, /\.\png$/],
  loader: require.resolve('url-loader'),
  options: {
    limit: 10000,
    name: 'static/media/[name].[hash:8].[ext]',
  },
},
// Process JS with Babel.
{
  test: /\.js|jsx$/,
  include: paths.appSrc,
  loader: require.resolve('babel-loader'),
  options: {
    compact: true,
  },
},
// "file" loader makes sure assets end up
// in the `build` folder.
// When you `import` an asset, you get
// its filename.
// This loader doesn't use a "test" so
// it will catch all modules
// that fall through the other loaders.
{
  loader: require.resolve('file-loader'),
  // Exclude `js` files to keep "css"
  // loader working as it injects
  // it's runtime that would otherwise
  // processed through "file" loader.
  // Also exclude `html` and `json`
  // extensions so they get processed
  // by webpack's internal loaders.
  exclude: [/\.\js$/, /\.\html$/, /\.\json$/],
  options: {
    name: 'static/media/[name].[hash:8].[ext]',
  },
},
```

```
  },
  test: /\.css$/,
  loader: ExtractTextPlugin.extract(
    Object.assign(
      {
        fallback: require.resolve('style-loader'),
        use: [
          {
            loader: require.resolve('css-loader'),
            options: {
              importLoaders: 1,
              minimize: true,
              sourceMap: shouldUseSourceMap,
            },
          },
          {
            loader: require.resolve('postcss-loader'),
            options: {
              // Necessary for external CSS imports to work
              ident: 'postcss',
              plugins: () => [
                require('postcss-flexbugs-fixes'),
                autoprefixer({
                  browsers: [
                    '>1%',
                    'last 4 versions',
                    'Firefox ESR',
                    'not ie < 9', // React doesn't support IE8 anyway
                  ],
                  flexbox: 'no-2009',
                }),
                ],
              },
            },
            ],
          },
        extractTextPluginOptions
      )
    ),
    // Note: this won't work without `new ExtractTextPlugin()` in `plugins`.
  ),
}
```

Plugins

- ❖ InterpolateHtmlPlugin
- ❖ HtmlWebpackPlugin
- ❖ DefinePlugin
- ❖ UglifyPlugin
- ❖ ExtractTextPlugin
- ❖ IgnorePlugin

```
plugins: [  
  // Makes some environment variables available in index.html.  
  // The public URL is available as %PUBLIC_URL% in index.html, e.g.:  
  // <link rel="shortcut icon" href="%PUBLIC_URL%/favicon.ico">  
  // In production, it will be an empty string unless you specify "homepage"  
  // in `package.json`, in which case it will be the pathname of that URL.  
  new InterpolateHtmlPlugin(env.raw),  
  // Generates an `index.html` file with the <script> injected.  
  new HtmlWebpackPlugin({  
    inject: true,  
    template: paths.appHtml,  
    minify: {  
      removeComments: true,  
      collapseWhitespace: true,  
      removeRedundantAttributes: true,  
      useShortDoctype: true,  
      removeEmptyAttributes: true,  
      removeStyleLinkTypeAttributes: true,  
      keepClosingSlash: true,  
      minifyJS: true,  
      minifyCSS: true,  
      minifyURLs: true,  
    },  
  }),  
  // Makes some environment variables available to the JS code, for example:  
  // if (process.env.NODE_ENV === 'production') { ... }. See `./env.js`.  
  // It is absolutely essential that NODE_ENV was set to production here.  
  // Otherwise React will be compiled in the very slow development mode.  
  new webpack.DefinePlugin(env.stringified),  
  // Minify the code.  
  new webpack.optimize.UglifyJsPlugin({  
    compress: {  
      warnings: false,  
      // Disabled because of an issue with Uglify breaking seemingly valid code:  
      // Pending further investigation:  
      // https://github.com/mishoo/UglifyJS2/issues/2011  
      comparisons: false,  
    },  
    output: {  
      comments: false,  
      // Turned on because emoji and regex is not minified properly using default  
      ascii_only: true,  
    },  
    sourceMap: shouldUseSourceMap,  
  }),  
  // Note: this won't work without ExtractTextPlugin.extract(..) in `loaders`.  
  new ExtractTextPlugin({  
    filename: cssFilename,  
  }),
```

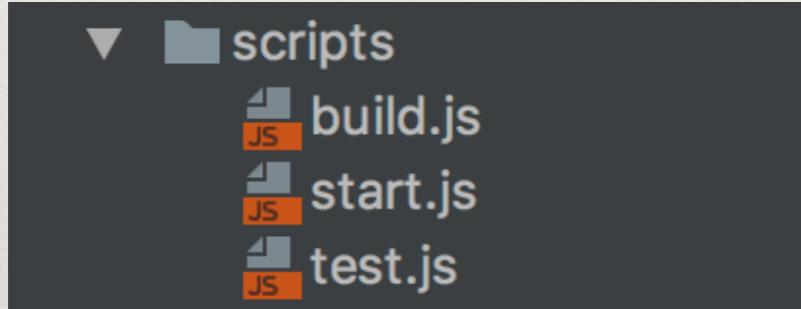
More Plugins

- ❖ FaviconsWebpackPlugin
- ❖ CommonsChunkPlugin
- ❖ ManifestPlugin
- ❖ SWrecacheWepackPlugin

```
new FaviconsWebpackPlugin({
  logo: paths.favicon,
  prefix: 'icons-[hash]/',
  emitStats: false,
  statsFilename: 'iconstats-[hash].json',
  persistentCache: true,
  inject: true,
  title: 'React',
  icons: {
    android: true,
    appleIcon: true,
    appleStartup: true,
    favicons: true,
    firefox: true,
    opengraph: false,
    twitter: false,
    windows: false
  }
}),
new webpack.optimize.CommonsChunkPlugin({
  name: 'vendor',
  filename: 'static/js/[name]-[hash:8].min.js',
  minChunks: (module) => module.context &&
  | module.context.indexOf("node_modules") !== -1,
}),
new SWPrecacheWebpackPlugin({
  // By default, a cache-busting query parameter is appended to requests
  // used to populate the caches, to ensure the responses are fresh.
  // If a URL is already hashed by Webpack, then there is no concern
  // about it being stale, and the cache-busting can be skipped.
  dontCacheBustUrlsMatching: /\.\\w{8}\\./,
  filename: 'service-worker.js',
  logger(message) { ... },
  minify: true,
  // For unknown URLs, fallback to the index page
  navigateFallback: publicUrl + '/index.html',
  // Ignores URLs starting from /__ (useful for Firebase):
  navigateFallbackWhitelist: [/^(?!\\__).*/],
  // Don't precache sourcemaps (they're large) and build asset manifest:
  staticFileGlobsIgnorePatterns: [/\\.map$/, /asset-manifest\\.json$/],
}),
```

Scripts

- ❖ start
- ❖ build
- ❖ staging
- ❖ test



```
"scripts": {  
  "start": "node scripts/start.js",  
  "build": "node scripts/build.js",  
  "test": "node scripts/test.js --env=jsdom"  
},
```

```
'use strict';

// Do this as the first thing so that any code reading it knows the right env.
process.env.BABEL_ENV = 'production';
process.env.NODE_ENV = 'production';

// Makes the script crash on unhandled rejections instead of silently
// ignoring them. In the future, promise rejections that are not handled will
// terminate the Node.js process with a non-zero exit code.
process.on('unhandledRejection', err => [...]);

// Ensure environment variables are read.
require('../config/env');

const path = require('path');
const chalk = require('chalk');
const fs = require('fs-extra');
const webpack = require('webpack');
const config = require('../config/webpack.config.prod');
const paths = require('../config/paths');
const checkRequiredFiles = require('react-dev-utils/checkRequiredFiles');
const formatWebpackMessages = require('react-dev-utils/formatWebpackMessages');
const printHostingInstructions = require('react-dev-utils/printHostingInstructions');
const FileSizeReporter = require('react-dev-utils/FileSizeReporter');
const printBuildError = require('react-dev-utils/printBuildError');

const measureFileSizesBeforeBuild =
  FileSizeReporter.measureFileSizesBeforeBuild;
const printFileSizesAfterBuild = FileSizeReporter.printFileSizesAfterBuild;
const useYarn = fs.existsSync(paths.yarnLockFile);

// These sizes are pretty large. We'll warn for bundles exceeding them.
const WARN_AFTER_BUNDLE_GZIP_SIZE = 512 * 1024;
const WARN_AFTER_CHUNK_GZIP_SIZE = 1024 * 1024;

// Warn and crash if required files are missing
if (!checkRequiredFiles([paths.appHtml, paths.appIndexJs])) {
  process.exit(1);
}

// First, read the current file sizes in build directory.
// This lets us display how much they changed later.
measureFileSizesBeforeBuild(paths.appBuild)
  .then(previousFileSizes => [...])
  .then(({ stats, previousFileSizes, warnings }) => [...],
    err => [...]
  );

```

```
// Create the production build and print the deployment instructions.
function build(previousFileSizes) {
  console.log('Creating an optimized production build...');

  let compiler = webpack(config);
  return new Promise((resolve, reject) => {
    compiler.run((err, stats) => {
      if (err) {
        return reject(err);
      }
      const messages = formatWebpackMessages(stats.toJson({}, true));
      if (messages.errors.length) {
        // Only keep the first error. Others are often indicative
        // of the same problem, but confuse the reader with noise.
        if (messages.errors.length > 1) {
          messages.errors.length = 1;
        }
        return reject(new Error(messages.errors.join('\n\n')));
      }
      if (
        process.env.CI &&
        (typeof process.env.CI !== 'string' ||
         process.env.CI.toLowerCase() !== 'false') &&
        messages.warnings.length
      ) {
        console.log(
          chalk.yellow(
            '\nTreating warnings as errors because process.env.CI = true.\n' +
            'Most CI servers set it automatically.\n'
          )
        );
        return reject(new Error(messages.warnings.join('\n\n')));
      }
      return resolve({
        stats,
        previousFileSizes,
        warnings: messages.warnings,
      });
    });
  });
}

function copyPublicFolder() {
  fs.copySync(paths.appPublic, paths.appBuild, {
    dereference: true,
    filter: file => file !== paths.appHtml,
  });
}
```

Back-end

Scripts

❖ nodeModules

```
const nodeModules = {}
fs.readdirSync('node_modules')
  .filter(function (x) {
    return ['.bin'].indexOf(x) === -1
  })
  .forEach(function (mod) {
    nodeModules[mod] = 'commonjs ' + mod
  })

module.exports = {
  target: 'node',
  entry: [
    require.resolve('./polyfills'),
    paths.appIndexJs
  ],
  output: {
    path: paths.appBuild,
    filename: '[name].[chunkhash:8].js',
  },
  externals: nodeModules,
  resolve: {
    modules: ['node_modules', paths.appNodeModules].concat(
      process.env.NODE_PATH.split(path.delimiter)
        .filter(Boolean)
    ),
    extensions: ['.js', '.json'],
    alias: {},
    plugins: [
      new ModuleScopePlugin(paths.appSrc, [paths.appPackageJson]),
    ],
  },
}
```

❖ target

```
plugins: [
  new webpack.optimize.UglifyJsPlugin({
    compress: {
      warnings: false,
      comparisons: false,
    },
    output: {
      comments: false,
      ascii_only: true,
    },
    sourceMap: shouldUseSourceMap,
  }),
  new webpack.DefinePlugin(env.stringifyed),
  new webpack.IgnorePlugin(/^\\.\\\locale$/|/moment$/),
],
node: {
  console: false,
  global: false,
  process: false,
  Buffer: false,
  __filename: true,
  __dirname: true,
},
```

❖ node

Libs

- ❖ library
- ❖ libraryTarget
 - var -> global (default)
 - this
 - window
 - umd -> AMD o CommonsJS
- ❖ externals
- ❖ package.json

```
module.exports = {
  entry: [
    paths.appIndexJs
  ],
  output: {
    path: paths.appBuild,
    filename: 'svjs-webpack.js',
    library: 'svjsWebpack',
    libraryTarget: 'umd',
  },
  externals: {
    lodash: {
      commonjs: 'lodash',
      commonjs2: 'lodash',
      amd: 'lodash',
      root: '_'
    }
  },
}
```

```
"main": "dist/svjs-webpack.js",
"module": "src/index.js",
```

¿Es recomendado usar siempre webpack?

No

Herram./Nivel	Principiante	Intermedio	Experto
Simple	No	No	¿No?
Create Kit	No	Recomendable	Si
Fram / Library !(Create Kit)	Recomendable	Si	Si
Lib	Recomendable	Si	Si
NodeJS	Recomendable	Si	Si

Ventajas y Desventajas

Ventajas

- ❖ Todo el potencial de webpack y los scripts.
- ❖ Su configuración y facilidad de entornos.
- ❖ Gráficos de dependencias.
- ❖ Todo se hace por y en JS.
 - Framework / Library

Desventajas

- ❖ Todo se hace por y en JS
 - import css
 - import assets (img, icons,...)

¿Alguna
pregunta?



Hazlo o no lo hagas, pero no lo intentes...

Muti page app

```
entry: [
  paths.appIndexJs,
  path.dashboardIndex.js,
],
```

- ❖ Le estamos diciendo a webpack que nos gustaría 2 gráficos de dependencia separados.
- ❖ En una aplicación de varias páginas, el servidor buscará un nuevo documento HTML. La página vuelve a cargar este nuevo documento y los assets se vuelven a descargar.

Async Routes

```
Creating an optimized production build...
Compiled successfully.

File sizes after gzip:

197.18 KB  build/static/js/vendor-612754e6.min.js
124.93 KB  build/static/js/main.d7bed986.js
70.22 KB   build/static/js/2.0b5aaae3.chunk.js
67.02 KB   build/static/css/main.836ff273.css
63.17 KB   build/static/js/7.3961441d.chunk.js
55.98 KB   build/static/js/13.376a508c.chunk.js
46.59 KB   build/static/js/3.fdb1d245.chunk.js
34.5 KB    build/static/js/8.99f50da6.chunk.js
32.92 KB   build/static/js/9.86b03b70.chunk.js
32.8 KB    build/static/js/4.fad99ff6.chunk.js
27.33 KB   build/static/js/11.973a975e.chunk.js
25.18 KB   build/static/js/5.deaf0de6.chunk.js
24.78 KB   build/static/js/6.faaf4aa3.chunk.js
21.56 KB   build/static/js/12.13013727.chunk.js
17.98 KB   build/static/js/14.6d41ac94.chunk.js
14.06 KB   build/static/js/10.65d70f31.chunk.js
12.38 KB   build/static/js/15.44db7480.chunk.js
11.58 KB   build/static/css/vendor.f0f37e5b.css
1.8 KB     build/static/js/20.d81919fb.chunk.js
1.75 KB   build/static/js/19.40ae411f.chunk.js
1.5 KB    build/static/js/17.2ce18351.chunk.js
1.47 KB   build/static/js/18.ded6f10f.chunk.js
1.39 KB   build/static/js/16.43706c45.chunk.js
1.1 KB    build/static/js/manifest.68f1e9db.js
917 B     build/static/js/30.5a174c10.chunk.js
895 B     build/static/js/21.9f559a70.chunk.js
861 B     build/static/js/29.c32ba9c4.chunk.js
746 B     build/static/js/31.1626529f.chunk.js
730 B     build/static/js/34.c7f31901.chunk.js
692 B     build/static/js/23.5991ea50.chunk.js
677 B     build/static/js/1.b9d2c9f7.chunk.js
659 B     build/static/js/0.dda18961.chunk.js
640 B     build/static/js/32.7a997ecc.chunk.js
563 B     build/static/js/25.71005a4e.chunk.js
530 B     build/static/js/35.60a3c605.chunk.js
504 B     build/static/js/27.b4caa2ff.chunk.js
500 B     build/static/js/22.e617e85c.chunk.js
444 B     build/static/js/24.7f536253.chunk.js
442 B     build/static/js/33.808423e9.chunk.js
386 B     build/static/js/28.7e19c553.chunk.js
254 B     build/static/js/26.8fb28127.chunk.js
```

Por fin, ahora
si...

