Examples

Table of Contents

Aggressive Merging

[aggressive-merging](#)

Chunk

[chunkhash](#)

[common-chunk-and-vendor-chunk](#)

[explicit-vendor-chunk](#)

[extra-async-chunk-advanced](#)

[extra-async-chunk](#)

[code-splitting-specify-chunk-name](#)

[named-chunks](#) example demonstrating merging of chunks with named chunks

[two-explicit-vendor-chunks](#)

Code Splitted

code-splitted-css-bundle

code-splitted-require.context-amd example demonstrating contexts in a code-split environment with AMD.

code-splitted-require.context example demonstrating contexts in a code-split environment.

Code Splitting

code-splitting example demonstrating a very simple case of Code Splitting.

code-splitting-bundle-loader example demonstrating Code Splitting through the builder loader

code-splitting-harmony

code-splitting-native-import-context

code-splitting-specify-chunk-name

Coffee Script

coffee-script example demonstrating code written in coffee-script.

CommonJS

commonjs example demonstrating a very simple program

DLL

[dll](dll)

dll-user

Externals

externals

Harmony

harmony

code-splitting-harmony

harmony-interop

harmony-library

harmony-unused

HTTP2 Aggressive Splitting

http2-aggressive-splitting

Hybrid Routing

hybrid-routing

Loader

loader example demonstrating the usage of loaders.

Mixed

mixed example demonstrating mixing CommonJs and AMD

Multi Compiler

multi-compiler

Multi Part Library

multi-part-library

Multiple Entry Points

multiple-entry-points example demonstrating multiple entry points with Code Splitting.

Require Context

require.context example demonstrating automatic creation of contexts when using variables in `require`.

Require Resolve

require.resolve example demonstrating how to cache clearing of modules with `require.resolve` and `require.cache`.

Scope Hoisting

scope-hoisting

Side Effects

side-effects

TypeScript

TypeScript

Source Map

source-map

WebAssembly

[wasm-simple](#) example demonstrating simple import from a WebAssembly module [wasm-complex](#) example demonstrating top-level await and import of WebAssembly text format with wast-loader

Web Worker

[web-worker](#) example demonstrating creating WebWorkers with webpack.

Requests

If you think an example is missing, please report it as issue. :)

Building an Example

1. Run `yarn` in the root of the project.
2. Run `yarn setup` in the root of the project.
3. Run `yarn add --dev webpack-cli` in the root of the project.
4. Run `node build.js` in the specific example directory. (Ex: `cd examples/commonjs && node build.js`)

Note: To build all examples run `npm run build:examples`