

This example show how to use Code Splitting with the ES6 module syntax.

The standard `import` is sync.

`import(module: string) -> Promise` can be used to load modules on demand. This acts as a split point for webpack and creates a chunk.

Providing dynamic expressions to `import` is possible. The same limits as with dynamic expressions in `require` calls apply here. Each possible module creates an additional chunk. In this example `import("c/" + name)` creates two additional chunks (one for each file in `node_modules/c/`). This is called "async context".

example.js

```
_({{example.js}})_
```

dist/output.js

```
_({{dist/output.js}})_
```

Info

Unoptimized

```
_({{stdout}})_
```

Production mode

```
_({{production:stdout}})_
```