This example demonstrates how webpack tracks the usage of ES6 imports and exports. Only used exports are emitted to the resulting bundle. The minimizing step then removes the declarations because they are unused.

Excluding unused exports from bundles is known as "tree-shaking".

In this example, only `add` and `multiply` in `./math.js` are used by the app. `list` is unused and is not included in the minimized bundle (Look for `Array.from` in the minimized bundle).

In addition to that, `library.js` simulates an entry point to a big library. `library.js` re-exports multiple identifiers from submodules. Often big parts of that are unused, like `abc.js`. Note how the usage information flows from `example.js` through `library.js` into `abc.js` and all declarations in `abc.js` are not included in the minimized bundle (Look for `console.log("a")` in the minimized bundle).

example.js

```
_{{example.js}}_
```

math.js

```
_{{math.js}}_
```

library.js

```
_{{library.js}}_
```

dist/output.js

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

dist/output.js

```
_{{production:dist/output.js}}_
```

Info

Unoptimized

```
_{{stdout}}_
```

Production mode

```
_{{production:stdout}}_
```