This example shows the automatically created async commons chunks.

The example entry references two chunks:

- entry chunk
  - async require -> chunk X
  - async require -> chunk Y
- chunk X
  - module `a`
  - module `b`
  - module `c`
- chunk Y
  - module `a`
  - module `b`
  - module `d`

These chunks share modules `a` and `b`. The optimization extract these into chunk Z:

Note: The optimization compares the size of chunk Z to some minimum value, but this is disabled from this example. In practice, there is no configuration needed for this.

- entry chunk
  - async require -> chunk X & Z
  - async require -> chunk Y & Z
- chunk X
  - module `c`
- chunk Y
  - module `d`
- chunk Z
  - module `a`
  - module `b`

Pretty useful for a router in a SPA.

example.js

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

dist/output.js

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

dist/394.output.js

```
_{{dist/394.output.js}}_
```

dist/460.output.js

```
_{{dist/460.output.js}}_
```

dist/767.output.js

```
_{{dist/767.output.js}}_
```

Info

Unoptimized

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

Production mode

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