

# Barrel Optimisation Script

This document lists the development and updates done to the barrel optimisation script:

- **AST Analysis of the exports:** The previous script failed to include dependencies of the exports. For eg:

```
// index.js
function Hello() {
  console.log('hello')
}

export const x = Hello();
```

This only added `export const x = Hello()` to the new Module and excluded `Hello` altogether. I have made updates to the code to do AST analysis of the exports and get their dependents and recursively do the same for each dependent, Thus ensuring that the newly formed Modules contain all the required code for the export.

- **Recursive Imports:** The previous script didn't consider recursive barrel exports

```
//index2.js
export * from './test' // test exports {test}

//index.js
export * from './index2.js'

//main.js
import {test} from './index.js'

// This was not mapped to './test' earlier
```

- **Traversal of the directory:** The previous script had a major issue in traversing the entry directory, resulting in many errors in cases of nested index files. It dealt with eliminating barrel exports and resolving imports in the same script which is not possible.

```
- Entry Directory
  - index.js
  - sub Directory
    - index.js
```

```
// Here the earlier script failed since for each index file, the
// the entire parent directory again
```

- **Circular dependency:** There were issues with circular dependencies occurring in the code splitted with the previous script.

```
// x.js
import {y} from './y.js'

... code

// y.js
import {x} from './x.js'

... code
```

- **Wildcard Imports solved:** The script had issues optimising wildcard imports. The introduced updates makes it possible to resolve such imports in a way that only changes the ImportDeclaration and the code remains functional:

```
// index.js
export { comp1 } from './test1'
export { comp2 } from './test2'

// page.js
```

```
import * as WildcardImport from './index'
// Modified to
import { comp1 } from './test1'
import { comp2 } from './test2'
const WildcardImport = {
  comp1: comp1,
  comp2: comp2
}
```

- **Default Exports from index file:** The default exports from index file gave and empty export error previously. Added resolution for such default exports as well.

```
// index.js
export { comp1 } from './test1'
export default function Check() {
  return (...)
}
// The export Check doesnt give an error now
```

- **Re-exports with Aliases:** Re-exports in the index file with aliases weren't mapped properly in the earlier work. So i added a map to store the local and exported names and used it to resolve such alias exports.

```
// index.js
export {default as MainComp} from './test'
export { Test1 as Renamed } from './test'

// page.js
import {MainComp} from './index'
// Modified to: import {default as MainComp} from './test'

import {Renamed} from './index'
// Modified to: import {Test1 as Renamed } from './test'
```