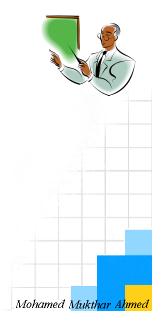


12

Custom Modules



Outline
Need for Modules
Expose Functionality
Importing CommonJS Modules
Ways of Exposing Objects



13





- Not a GOOD IDEA to have everything (data and functionality) in one single Node.js file
- Moreover, as the application become bigger in size, we have concerns
- CommonJS modules are the original way to package JavaScript code for Node.js.
- Node.js also supports the ECMAScript modules standard used by browsers and other JavaScript runtimes.
- Every Node.js file is a module (by default)



14

Expose Functionality



- Any object or variable defined in the file by default is private and not exposed to the outer world.
- The **module.exports** API offered by the module system allows us to expose objects.

```
const {PI} = Math

const area = (r) => { return PI * r ** 2 };

const circumference = (r) => { return 2 * PI * r };

module.exports = {area, circumference};
```

Mohamed Mukthar Ahmed

15

Importing a CommonJS Module de la commonJS Module

- A Node.js file can import components exposed by other Node.js files
- When you want to import a CommonJS module use require()

```
const circle = require('./circle');
console.log(`Area of circle with radius 4 is: ${circle.area(4)}`);
console.log(`Circumference of circle with radius 4 is: ${circle.circumference(4)}`);

Mohamed Mukthar Ahmed
```

16

Ways of Exposing Objects



- Two ways to expose objects
 - Using module.exports
 - Using exports
- Difference
 - With module.exports the objects are exposed
 - With exports the properties of the object are exposed

Mohamed Mukth<mark>ar Ahmed</mark>

17