



Node.js Custom Modules



Mohamed Mukthar Ahmed

12



Custom Modules

Outline

Need for Modules

Expose Functionality

Importing CommonJS Modules

Ways of Exposing Objects



Mohamed Mukthar Ahmed

13

Need for Modules



- 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)

Mohamed Mukhtar Ahmed

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.

```
1  const {PI} = Math
2
3  const area = (r) => { return PI * r ** 2 };
4  const circumference = (r) => { return 2 * PI * r };
5
6  module.exports = {area, circumference};
```

Mohamed Mukhtar Ahmed

15

■ Importing a 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()**

```
1 const circle = require('./circle');
2
3 console.log('Area of circle with radius 4 is: ${circle.area(4)}');
4 console.log('Circumference of circle with radius 4 is: ${circle.circumference(4)}');
5
```

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

```
1 module.exports = {
2   brand: 'Ford',
3   model: 'Figo'
4 };
```

```
1 exports.car = {
2   brand: 'Maruti',
3   model: 'Swift'
4 };
```

Mohamed Mukthar Ahmed

17