Lecture-15

Middlewares

Coding Blocks - Kartik Mathur

Class Agenda

01	Middlewares		
02	Module-Exports		
03	Todo app	04	-
05	-		
06	-	07	-

Middlewares

1

Middlewares

An Express application can use the following types of middleware:

- Application-level middleware
- Built-in middleware
- Third-party middleware
- Error-handling middleware
- Router-level middleware: We will see this later on

Middlewares

1. It is a generic middleware it runs for every request that comes to the express server.

```
app.use((req, res, next) => {
        console.log('Time:', Date.now())
        next()
})
```

2. It's another path specific middleware

```
app.use('/user/:id', (req, res, next) => {
        console.log('Request Type:', req.method)
        next()
})
```

3. Path fixed middlewares

```
app.get('/user/:id', (req, res) => {
        res.send('hello, user!')
})
```

Application level middlewares

Built in middleware

Express has the following built-in middleware functions:

- express.static serves static assets such as HTML files, images, and so on.
- express.json parses incoming requests with JSON payloads.
- express.urlencoded parses incoming requests with URL-encoded payloads.

Add them inside `app.use( express.json() );`

Middlewares

Error handling middleware

```
app.use((err, req, res, next) => {
        console.error(err)
        res.send('Something not right')
})
```

Middleware

Third party middleware

```
// load the body-parser middleware
npm i body-parser
```

```
// We can also use this instead of express.json() inbuilt middleware.
app.use(bodyparser.json())
```

HTTP Requests

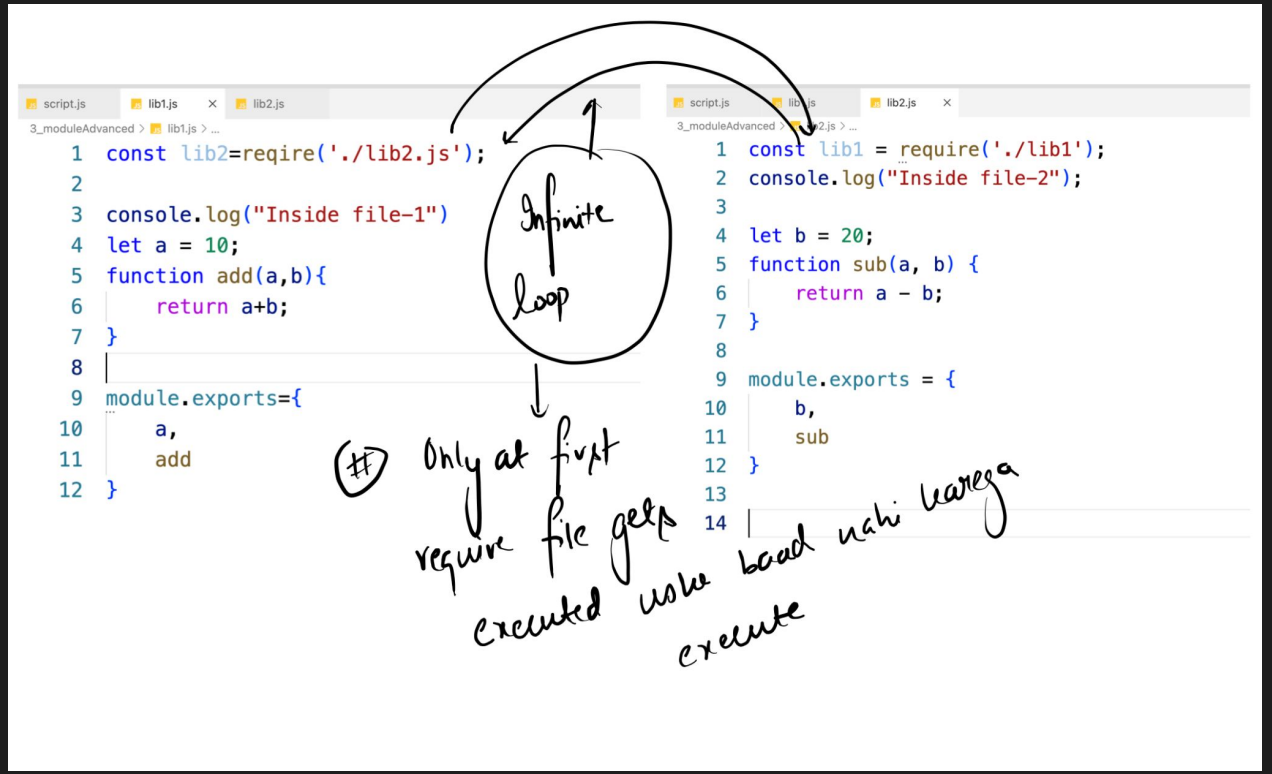
File-Dependency

2

Dependency Map

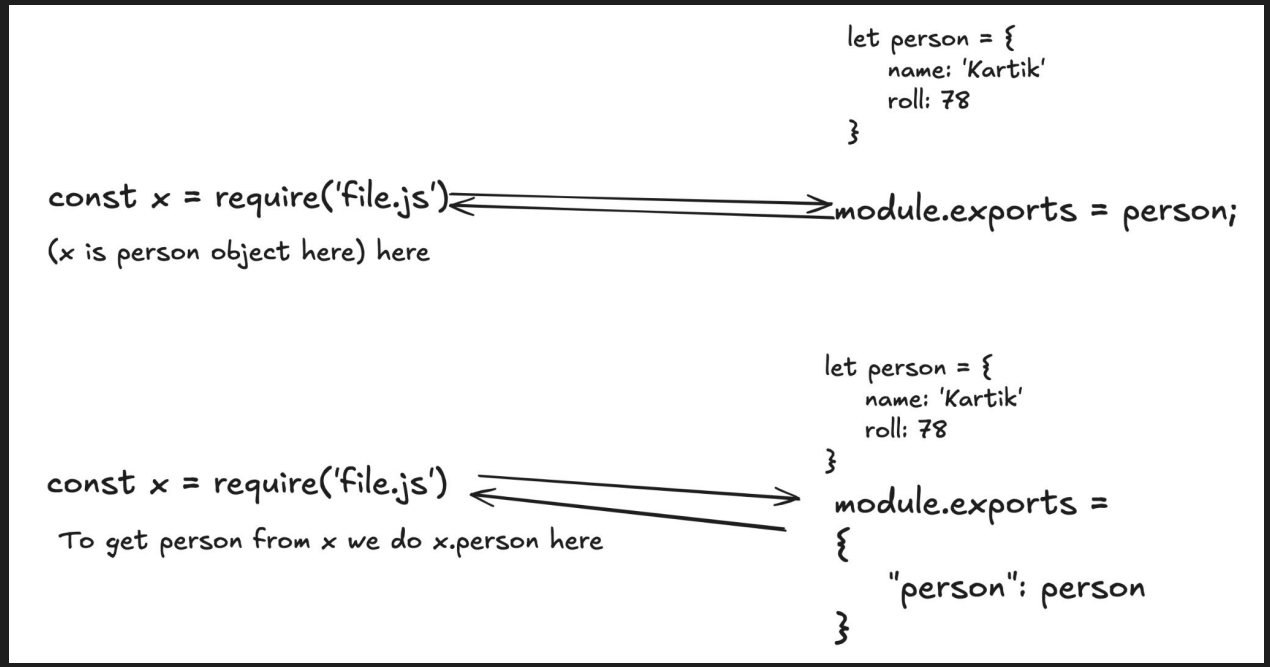
script.js

```
1  const lib1 = require('./lib1');
2  const lib2 = require('./lib2');
3
4  console.log(lib1);
5  console.log(lib2);
```



How it works?

Difference in module.exports



How it works?

Building TODO APP

3