

▼ ex2-ex3-ex6-ex7

▼ libraries

Model

JS calculator.js

JS idGenerator.js

JS logger.js

> node\_modules

▼ public

View

<> calculator.html

▼ routes

Controller

JS calculator.js

▼ tests

JS calculator.test.js

JS app.js

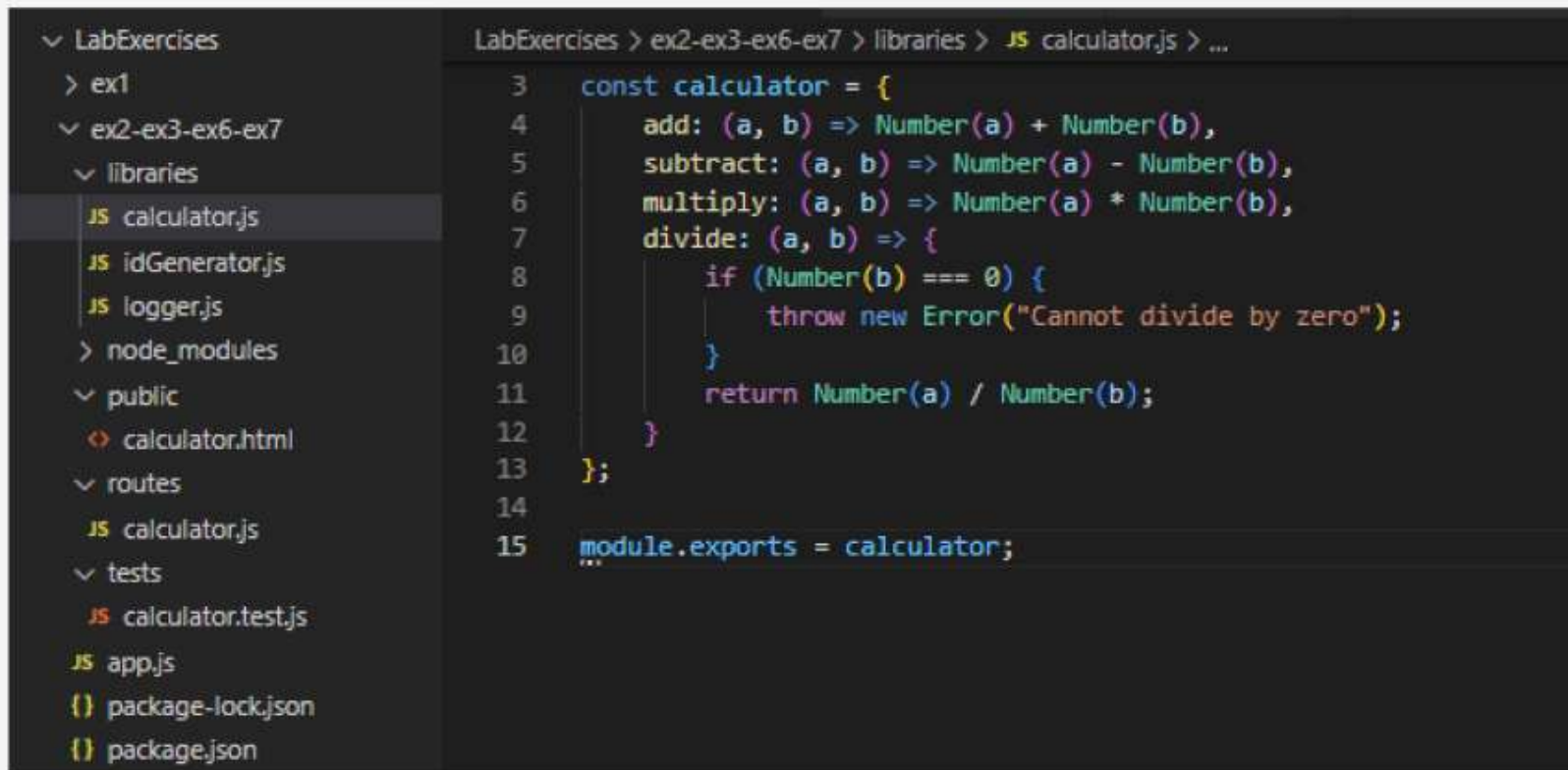
{ } package-lock.json

{ } package.json

# Model

(Business Logic / Data Layer)

libraries/calculator.js



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'LabExercises', 'ex1', 'ex2-ex3-ex6-ex7', and 'libraries'. The 'libraries' folder is expanded, showing 'calculator.js' selected. The code editor shows the content of 'calculator.js' with line numbers 3 to 15. The code defines a 'calculator' object with methods 'add', 'subtract', 'multiply', and 'divide'. The 'divide' method includes a check for division by zero. The object is then exported as 'module.exports'.

```
3  const calculator = {
4    add: (a, b) => Number(a) + Number(b),
5    subtract: (a, b) => Number(a) - Number(b),
6    multiply: (a, b) => Number(a) * Number(b),
7    divide: (a, b) => {
8      if (Number(b) === 0) {
9        throw new Error("Cannot divide by zero");
10     }
11     return Number(a) / Number(b);
12   }
13 };
14
15 module.exports = calculator;
```

- This contains the actual logic to perform calculations.
- It should contain pure functions like `add()`, `sub()`, `multiply()` & `divide()`.

```
LabExercises > ex2-ex3-ex6-ex7 > public > calculator.html > html > body
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Basic Calculator</title>
6   <style>
7     body {
8       font-family: Arial, sans-serif;
9       padding: 20px;
10    }
11    input, select, button {
12      padding: 10px;
13      margin: 5px;
14    }
15  </style>
16 </head>
17 <body>
18   <h2>Simple Calculator</h2>
19
20   <input type="number" id="num1" placeholder="Enter first number">
21   <input type="number" id="num2" placeholder="Enter second number">
22
23   <select id="operation">
24     <option value="add">Add</option>
25     <option value="subtract">Subtract</option>
26     <option value="multiply">Multiply</option>
27     <option value="divide">Divide</option>
28   </select>
29
30   <button onclick="calculate()">Calculate</button>
31
32   <h3>Result: <span id="result">---</span></h3>
33   You, 2 months ago • Init commit
34   <script>
35     function calculate() {
36       const a = document.getElementById("num1").value;
37       const b = document.getElementById("num2").value;
38       const operation = document.getElementById("operation").value;
39
40       if (a === '' || b === '') {
41         alert('Please enter both numbers!');
42         return;
43       }
44
45       fetch('http://localhost:3000/${operation}?a=${a}&b=${b}')
46         .then(response => response.json())
47         .then(data => {
48           document.getElementById("result").innerText = data.result ?? 'Error';
49         })
50         .catch(error => {
51           document.getElementById("result").innerText = 'Error';
52           console.error('Error:', error);
53         });
54     }
55   </script>
56 </body>
57 </html>
```

# View

(Frontend / UI)

public/calculator.html

- This is the user interface for input and output (e.g., HTML + CSS).
- UI to interact with the app.
- It should mainly contain the input form, not logic.



```
LabExercises > ex2-ex3-ex6-ex7 > routes > JS calculator.js > ...
You, 2 months ago | 1 author (You)
1  const express = require('express');
2  const router = express.Router();
3
4  const calculator = require('../libraries/calculator.js');
5  const generateRandomId = require('../libraries/idGenerator.js');
6  const logger = require('../libraries/logger.js');
7
8  // Add
9  router.get('/add', (req, res) => {
10     const { a, b } = req.query;
11     const result = calculator.add(a, b);
12     const id = generateRandomId();
13     logger(id, 'add', result);
14     res.json({ id, result });
15 });
16
17 // Subtract
18 router.get('/subtract', (req, res) => {
19     const { a, b } = req.query;
20     const result = calculator.subtract(a, b);
21     const id = generateRandomId();
22     logger(id, 'subtract', result);
23     res.json({ id, result });
24 });
25
26 // Multiply
27 router.get('/multiply', (req, res) => {
28     const { a, b } = req.query;
29     const result = calculator.multiply(a, b);
30     const id = generateRandomId();
31     logger(id, 'multiply', result);
32     res.json({ id, result });
33 });
34
35 // Divide
36 router.get('/divide', (req, res) => {
37     const { a, b } = req.query;
38     try {
39         const result = calculator.divide(a, b);
40         const id = generateRandomId();
41         logger(id, 'divide', result);
42         res.json({ id, result });
43     } catch (error) {
44         res.status(400).json({ error: error.message });
45     }
46 });
47
48 module.exports = router;
```

# Controller

(Handles Request & Response)

routes/calculator.js

- Connects user input to the logic and sends back the response.
- Handles inputs and invokes the model.
- It should handle incoming requests and call model.

# App

(Server Setup & Route Definition)

## app.js

```
LabExercises > ex2-ex3-ex6-ex7 > JS app.js > ...  
41 app.get('/multiply', (req, res) => {  
42   const { a, b } = req.query;  
43   const result = Number(a) * Number(b);  
44  
45   const id = Math.floor(Math.random() * 1000000);  
46   logMessage(id, 'multiply', result);  
47   res.json({ id, result });  
48 });  
49  
50 app.get('/divide', (req, res) => {  
51   const { a, b } = req.query;  
52   if (Number(b) === 0) return res.status(400).json({ error: "Cannot divide by zero" });  
53   const result = Number(a) / Number(b);  
54  
55   const id = Math.floor(Math.random() * 1000000);  
56   logMessage(id, 'divide', result);  
57   res.json({ id, result });  
58 });  
59  
60 // Server listening  
61 const PORT = 3000;  
62 app.listen(PORT, () => {  
63   console.log(`Server running at http://localhost:${PORT}`);  
64 });  
65  
66 module.exports = app;
```

- This is the entry point, connects everything and starts the server.
- It should import controller and serve frontend properly.