

## 4.2C: Enhanced Functionality for the Calculator Microservice (Zeyu Chen)

GitHub Link (private):

[https://github.com/Cikky0209/SIT323\\_ZeyuChen.git](https://github.com/Cikky0209/SIT323_ZeyuChen.git)

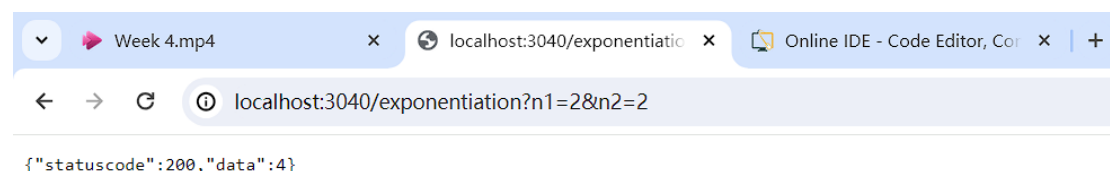
Submission:

### 1. Additional Arithmetic Operations:

- Exponentiation:

```
48 //Exponetiation
49 const exponetiation = (n1, n2) => {
50   return Math.pow(n1, n2);
51 }
52
```

```
157 // '^' function
158 app.get("/exponentiation", (req, res) => {
159   try {
160     const n1 = parseFloat(req.query.n1);
161     const n2 = parseFloat(req.query.n2);
162     if (isNaN(n1)) {
```



- **Square root:**

```
53 //Square root
54 const suquareRoot = (n) => {
55   return Math.sqrt(n);
56 }
57
```

```
180 // square root function
181 app.get("/square-root", (req, res) => {
182   try {
183     const n = parseFloat(req.query.n);
184     if (isNaN(n)) {
```

Week 4.mp4 x localhost:3040/square-root?n=121 x Online IDE - Code Editor, Cor x +

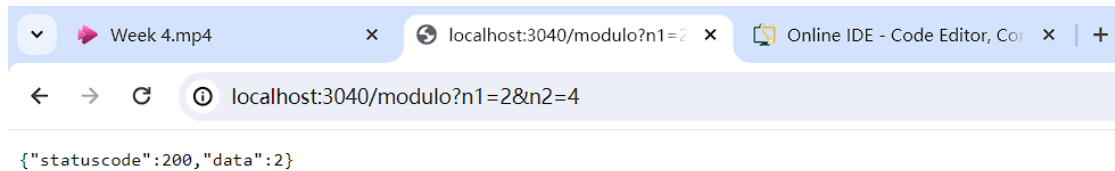
localhost:3040/square-root?n=121

```
{"statusCode":200,"data":11}
```

- **Modulo operations:**

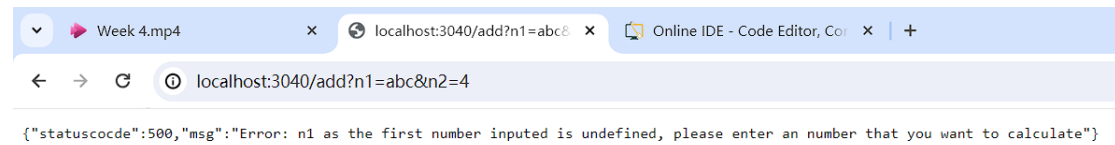
```
58 //Modulo operations
59 const modulo = (n1, n2) => {
60   |   return n1 % n2;
61 }
```

```
198 // '%' function
199 app.get("/modulo", (req, res) => {
200   |   try {
201   |       |   const n1 = parseFloat(req.query.n1);
202   |       |   const n2 = parseFloat(req.query.n2);
```

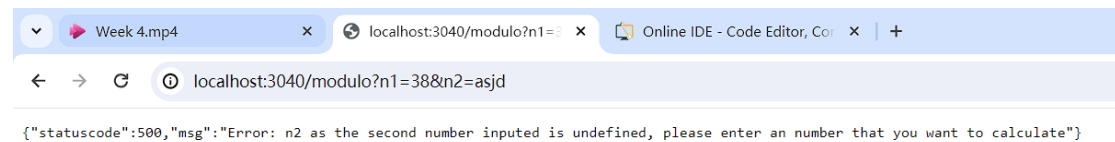


## 2. Error Handling:

- More detail when user input the content with n1 and n2.

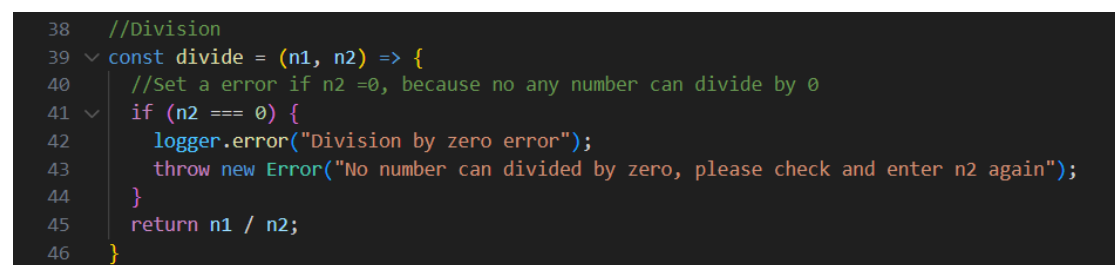


The screenshot shows a web browser with two tabs: 'Week 4.mp4' and 'localhost:3040/add?n1=abc&n2=4'. The address bar displays 'localhost:3040/add?n1=abc&n2=4'. The page content shows a JSON response: `{"statusCode":500,"msg":"Error: n1 as the first number inputed is undefined, please enter an number that you want to calculate"}`.

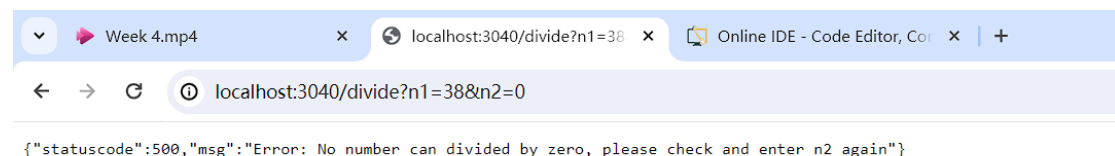


The screenshot shows a web browser with two tabs: 'Week 4.mp4' and 'localhost:3040/modulo?n1=38&n2=asjd'. The address bar displays 'localhost:3040/modulo?n1=38&n2=asjd'. The page content shows a JSON response: `{"statusCode":500,"msg":"Error: n2 as the second number inputed is undefined, please enter an number that you want to calculate"}`.

- Division by zero error.



```
38 //Division
39 const divide = (n1, n2) => {
40   //Set a error if n2 =0, because no any number can divide by 0
41   if (n2 === 0) {
42     logger.error("Division by zero error");
43     throw new Error("No number can divided by zero, please check and enter n2 again");
44   }
45   return n1 / n2;
46 }
```



The screenshot shows a web browser with two tabs: 'Week 4.mp4' and 'localhost:3040/divide?n1=38&n2=0'. The address bar displays 'localhost:3040/divide?n1=38&n2=0'. The page content shows a JSON response: `{"statusCode":500,"msg":"Error: No number can divided by zero, please check and enter n2 again"}`.