






```
{} package.json
```

node      ... ^ x

```
PS C:\Users\ggmed\Desktop\EstudosFlavio\task9> node index.js
Error loading data file. Starting with an empty array.
Servidor Funcionando!
```

## > LINHA DO TEMPO

## TASK9

&gt; node\_modules

JS index.js

{ } package-lockjson

{ } package.json

JS index.js &gt; [0] saveDataToFile

```
35
36   const numericHeight = parseFloat(height)
37   if (isNaN(numericHeight) || numericHeight < 50 || numericHeight > 250) {
38     return res.status(400).json({ error: 'Invalid height. It must be a number between 50 and 250.'})
39   } else {
40     next();
41   }
42 }
43 }
44
45 server.post('/persons', validatePerson, (req, res) => {
46   const newPerson = req.body;
47   persons.push(newPerson);
48   saveDataToFile();
49   res.status(201).json(newPerson);
50 });
51
52 server.get('/persons', (req, res) => {
53   res.json(persons);
54 });
55
56 server.get('/persons/:name', (req, res) => {
57   const { name } = req.params;
58   const person = persons.find((p) => p.name === name);
59
60   if (person) {
61     res.json(person);
62   } else {
63     res.status(404).json({ error: 'Person not found' });
64   }
65 });
66
67 server.put('/persons/:name', (req, res) => {
68   const { name } = req.params;
69   const updatedPerson = req.body;
70   const index = persons.findIndex((p) => p.name === name);
71
72   if (index !== -1) {
73     persons[index] = { ...persons[index], ...updatedPerson };
74     saveDataToFile();
75     res.json(persons[index]);
76   } else {
77     res.status(404).json({ error: 'Person not found' });
78   }
79 });
80
```

```
1 // ...
2 // ...
3 // ...
4 // ...
5 // ...
6 // ...
7 // ...
8 // ...
9 // ...
10 // ...
11 // ...
12 // ...
13 // ...
14 // ...
15 // ...
16 // ...
17 // ...
18 // ...
19 // ...
20 // ...
21 // ...
22 // ...
23 // ...
24 // ...
25 // ...
26 // ...
27 // ...
28 // ...
29 // ...
30 // ...
31 // ...
32 // ...
33 // ...
34 // ...
35 // ...
36 // ...
37 // ...
38 // ...
39 // ...
40 // ...
41 // ...
42 // ...
43 // ...
44 // ...
45 // ...
46 // ...
47 // ...
48 // ...
49 // ...
50 // ...
51 // ...
52 // ...
53 // ...
54 // ...
55 // ...
56 // ...
57 // ...
58 // ...
59 // ...
60 // ...
61 // ...
62 // ...
63 // ...
64 // ...
65 // ...
66 // ...
67 // ...
68 // ...
69 // ...
70 // ...
71 // ...
72 // ...
73 // ...
74 // ...
75 // ...
76 // ...
77 // ...
78 // ...
79 // ...
80 // ...
81 // ...
82 // ...
83 // ...
84 // ...
85 // ...
86 // ...
87 // ...
88 // ...
89 // ...
90 // ...
91 // ...
92 // ...
93 // ...
94 // ...
95 // ...
96 // ...
97 // ...
98 // ...
99 // ...
100 // ...
```

```
JS index.js > saveDataToFile  
80  
81 server.delete('/persons/:name', (req, res) => {  
82     const { name } = req.params;  
83     const index = persons.findIndex((p) => p.name === name);  
84  
85     if (index !== -1) {  
86         const deletedPerson = persons.splice(index, 1);  
87         saveDataToFile();  
88         res.json(deletedPerson);  
89     } else {  
90         res.status(404).json({ error: 'Person not found' });  
91     }  
92 });  
93  
94 server.listen(8888, () => {  
95     console.log('Servidor Funcionando!');  
96 });  
97
```

