

Conteúdo

Módulo 1: Introdução ao Node.JS

- Criando o editor um novo game

Agora iremos criar nossa rota para editarmos um game já existente em nosso array que está simulando um banco de dados.

Para isso precisaremos trabalhar com índices, que seriam a mesma situação em um banco de dados só que trabalhando com “id” o nosso índice funcionará da mesma forma que o “id”.

```
//att um curso
app.put('/novogame/:index', (req, res) =>{
  const { index } = req.params;
  let title = req.body.title;
  let studio = req.body.studio;
  let price = req.body.price;

  games[index] = { title, studio, price };

  return res.json(games);
})
```

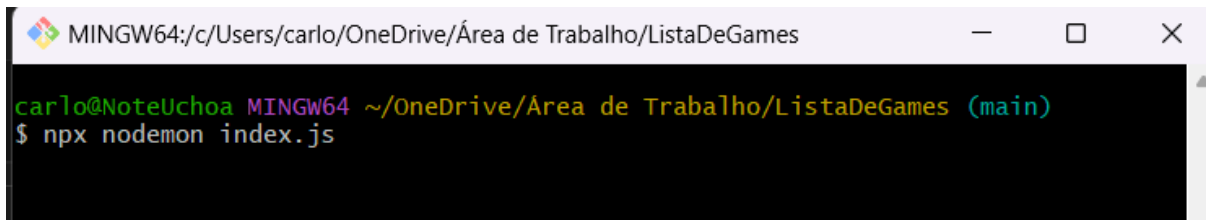
Como podem observar, agora temos um tipo de envio de informações do tipo **PUT** e uma rota nova “/novogame/:index” dessa forma eu consigo capturar qual jogo cadastrado eu pretendo realizar a edição, o nome **index** nesse quesito significa que pretendo passar um parâmetro no formato de índice de um array ok?

É preciso informar que irei trabalhar com índice e que irei passar por parâmetros os requisitos necessários para recuperar e enviar informações novas ok? Na sequência eu recupero as informações em variáveis como fizemos no processo de inserir dados , e depois chamo meu array games agora setando um novo índice para ele e por fim retorno o meu novo array com um produto modificado no formato json.

Feito isso podemos testar nossa API e verificar, vamos primeiro listar os games e verificarmos como estão listados.

Rodamos o servidor com o nodemon:

Aqui no exemplo usei o cmd com o meu git bash para ter maior visibilidade em tela das informações.



```
MINGW64:/c/Users/carlo/OneDrive/Área de Trabalho/ListaDeGames
carlo@NoteUchoa MINGW64 ~/OneDrive/Área de Trabalho/ListaDeGames (main)
$ npx nodemon index.js
```

Ao fazermos isso iremos agora listar nosso array de games:

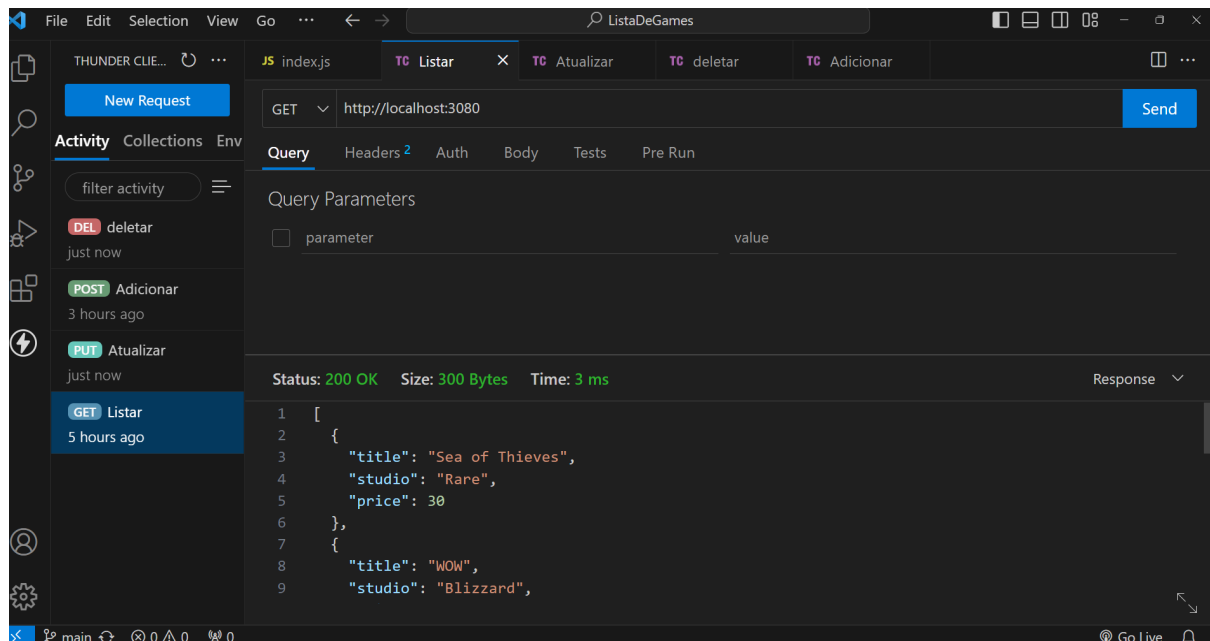
Podemos ver em nosso navegador pois a rota é do tipo **GET** e conseguimos acessar ou usar o Thunder Client pelo Vscode também:



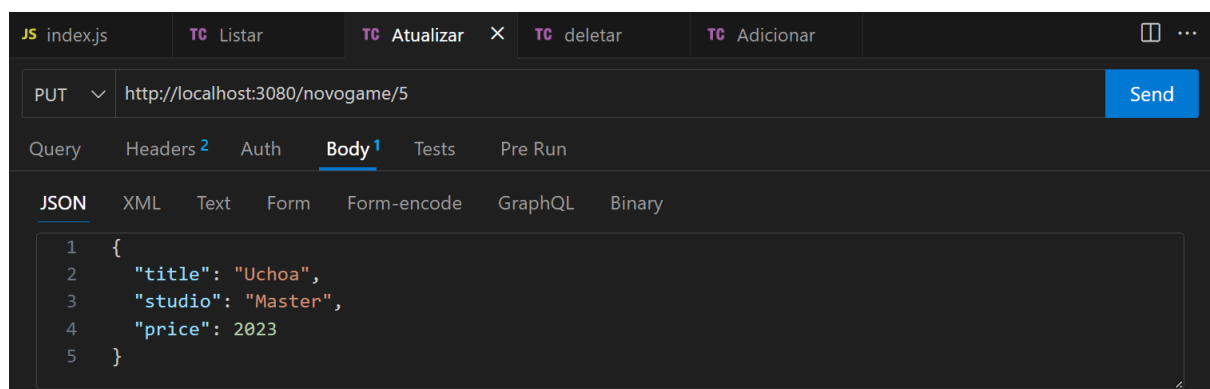
```
localhost:3080
Aplicar estilos de formatação ✓

[
  {
    "title": "Sea of Thieves",
    "studio": "Rare",
    "price": 30
  },
  {
    "title": "WOW",
    "studio": "Blizzard",
    "price": 120
  },
  {
    "title": "Valorant",
    "studio": "Riot",
    "price": 0
  },
  {
    "title": "COD",
    "studio": "Activision",
    "price": 200
  },
  {
    "title": "Minecraft",
    "studio": "Mojang",
    "price": 80
  },
  {
    "title": "Halo",
    "studio": "Microsoft",
    "price": 90
  }
]
```

Ou pelo Thunder Client:



Temos a quantidade e noção de índices que possuem o nosso array totalizando 5 posições, [0, 1, 2, 3, 4, 5], cada posição é um jogo cadastrado. Com essa informação em nossas mãos então vamos atualizar um destes jogos. Vamos criar uma nossa requisição no Thunder Client do tipo **PUT** e colocar o seguinte endereço: ["http://localhost:3080/novogame/5"](http://localhost:3080/novogame/5) lembrem que precisávamos passar um parâmetro do tipo índice? e ainda precisamos preencher o corpo do documento para enviarmos que é no formato **Json conforme imagem abaixo**, feito estas alterações basta clicarmos no botão enviar, notem que estamos alterando o jogo que está no índice de número **5** e estamos alterando o valor do título do jogo para o meu nome **Uchoa**, o studio para o nome **Master** e o price para o número **2023**, com todas as informações preenchidas, vamos clicar em enviar, no botão **(Send)** e após isso iremos listar os jogos novamente.



UniSENAI

Ao clicarmos no botão (**Send**):

O mesmo ja nos retorna o resultado :

The screenshot shows a REST client interface with a dark theme. At the top, there are tabs for 'JS index.js', 'TC Listar', 'TC Atualizar', and 'TC'. Below these, a dropdown menu shows 'PUT' and a text field contains the URL 'http://localhost:3080/novogame/5'. The main area has tabs for 'Query', 'Headers 2', 'Auth', 'Body 1', 'Tests', and 'Pre Run'. The 'Body 1' tab is selected, and within it, the 'JSON' sub-tab is active. The JSON body is displayed with line numbers 1 through 5, showing an object with 'title', 'studio', and 'price' properties. Below the body, the status bar indicates 'Status: 200 OK', 'Size: 300 Bytes', and 'Time: 12 ms'. The response body is shown at the bottom with line numbers 24 through 32, displaying a JSON array containing two objects.

```
1  {
2    "title": "Uchoa",
3    "studio": "Master",
4    "price": 2023
5  }
```

Status: 200 OK Size: 300 Bytes Time: 12 ms

```
24    "studio": "Mojang",
25    "price": 80
26  },
27  {
28    "title": "Uchoa",
29    "studio": "Master",
30    "price": 2023
31  }
32  ]
```

Podemos listar novamente para vermos no navegador também:



A screenshot of a web browser window. The address bar shows 'localhost:3080'. Below the address bar, there is a checkbox labeled 'Aplicar estilos de formatação' which is checked. The main content area displays a JSON array of game objects. Each object has 'title', 'studio', and 'price' fields. The games listed are 'Sea of Thieves', 'WOW', 'Valorant', 'COD', 'Minecraft', and 'Uchoa'.

```
[
  {
    "title": "Sea of Thieves",
    "studio": "Rare",
    "price": 30
  },
  {
    "title": "WOW",
    "studio": "Blizzard",
    "price": 120
  },
  {
    "title": "Valorant",
    "studio": "Riot",
    "price": 0
  },
  {
    "title": "COD",
    "studio": "Activision",
    "price": 200
  },
  {
    "title": "Minecraft",
    "studio": "Mojang",
    "price": 80
  },
  {
    "title": "Uchoa",
    "studio": "Master",
    "price": 2023
  }
]
```

Dessa forma concluímos nossa OT, e aprendemos a editar um tipo de dado inserido anteriormente via API, aguardo vocês nas próximas aulas.

Mas antes de terminarmos, agora vem a atividade que vocês tanto adoram.

ATIVIDADE

Quero que vocês realizem a edição em 5 jogos dos 10 que pedi anteriormente para cadastrarem com nome de outros jogos, e me enviem com essa alteração.