ESCOLA SENAI DE ITU
CENTRO EDUCACIONAL 401
ENSINO MÉDIO INTEGRADO AO TÉCNICO

HELOÍSA GABRIELLY PAIXÃO

# **RELATÓRIO**: MODELAGEM E MANIPULAÇÃO DE DADOS

SALTO
2025

# Rotas e Funcionalidades Completas

```
import express from "express"

const app = express()

app.use(express.json())

const livros = [
    {
        isbn: 1,
        titulo_livro: "Java - Como programar",
        editora: "Saber",
        ano_publicacao: "2002"
    },
    {
        isbn: 2,
        titulo_livro: "Java - Como programar de verdade",
        editora: "Saber",
        ano_publicacao: "2008"
    }
]


const autor = [
    {
        id_autor: 1,
        nome_autor: "Machado de Assis",
        nacionalidade: "Brasileiro"
    },
    {
        id_autor: 2,
        nome_autor: "Clarice Lispector",
        nacionalidade: "Brasileira"
    }
]


const autorLivro = [
    {
        id_autorLivro: 1,
        isbn: 1,
        id_autor: 1
```

```
    },
    {
        id_autor: 2,
        isbn: 2,
        id_autor: 2
    }
]


const exemplar = [
    {
        id_exemplar: 1,
        status_exemplar: "Disponível",
        isbn: 1
    },
    {
        id_exemplar: 2,
        status_exemplar: "Emprestado",
        isbn: 2
    }
]


const membro = [
    {
        id_membro: 1,
        nome_membro: "Letícia Roberta",
        endereco: "Rua das Flores 191 Bela Vista",
        telefone: "(11)98765-4321"
    },
    {
        id_membro: 2,
        nome_membro: "Mônica Cotrim",
        endereco: "Rua dos Tamanduá Bandeira 30, Amazona",
        telefone: "(11)91234-5678"
    }
]


const emprestimo = [
    {
        id_emprestimo: 1,
        data_emprestimo: "2025-11-24",
```

```
        data_devolucao: "2025-12-02",
        data_devolucao_efetiva: null,
        id_exemplar: 2,
        id_membro: 2
    },
    {
        id_emprestimo: 2,
        data_emprestimo: "2025-11-10",
        data_devolucao: "2025-11-24",
        data_devolucao_efetiva: "2025-11-17",
        id_exemplar: 1,
        id_membro: 1
    }
]

////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////

function buscarLivro(isbn) {
    return livros.findIndex(livro => {
        return livro.isbn === Number(isbn)
    })
}

app.get("/", (req, res) => {
    res.status(200).send("Livraria Saber e Cia")
})

app.get("/livros", (req, res) => {
    res.status(200).json(livros)
})

app.get("/livros/:isbn", (req, res) => {
    const index = buscarLivro(req.params.isbn)
    res.status(200).json(livros[index])
})

app.post("/livros", (req, res) => {
    livros.push(req.body)
    res.status(201).json(req.body)
})

app.put("/livros/:isbn", (req, res) => {
```

```
    const index = buscarLivro(req.params.isbn)

    livros[index].titulo_livro = req.body.titulo_livro
    livros[index].editora = req.body.editora
    livros[index].ano_publicacao = req.body.ano_publicacao

    res.status(200).json(livros[index])
})

app.delete("/livros/:isbn", (req, res) => {
    const index = buscarLivro(req.params.isbn)
    livros.splice(index, 1)
    res.status(200).json(livros)
});

////////////////////////////////////////////////////////////////////////
////////////////////////////////////////

function buscarAutor(id_autor) {
    return autor.findIndex(autor => {
        return autor.id_autor === Number(id_autor)
    })
}

app.get("/", (req, res) => {
    res.status(200).send("Livraria Saber e Cia")
})

app.get("/autor", (req, res) => {
    res.status(200).json(autor)
})

app.get("/autor/:id_autor", (req, res) => {
    const index = buscarAutor(req.params.id_autor)
    res.status(200).json(autor[index])
})

app.post("/autor", (req, res) => {
    autor.push(req.body)
    res.status(201).json(req.body)
})

app.put("/autor/:id_autor", (req, res) => {
```

```
    const index = buscarAutor(req.params.id_autor)

    autor[index].id_autor = req.body.id_autor
    autor[index].nome_autor = req.body.nome_autor
    autor[index].nacionalidade = req.body.nacionalidade

    res.status(200).json(autor[index])
})

app.delete("/autor/:id_autor", (req, res) => {
    const index = buscarAutor(req.params.id_autor)
    autor.splice(index, 1)
    res.status(200).json(autor)
});

////////////////////////////////////////////////////////////////////////
////////////////////////////////////////

function buscarAutorLivro(isbn) {
    return autorLivro.findIndex(autorLivro => {
        return autorLivro.isbn === Number(isbn)
    })
}

app.get("/", (req, res) => {
    res.status(200).send("Livraria Saber e Cia")
})

app.get("/autorLivro", (req, res) => {
    res.status(200).json(autorLivro)
})

app.get("/autorLivro/:isbn", (req, res) => {
    const index = buscarAutorLivro(req.params.isbn)
    res.status(200).json(autorLivro[index])
})

app.post("/autorLivro", (req, res) => {
    autorLivro.push(req.body)
    res.status(201).json(req.body)
})

app.put("/autorLivro/:isbn", (req, res) => {
```

```
    const index = buscarAutorLivro(req.params.isbn)

    autor[index].id_autorLivro = req.body.id_autorLivro
    autor[index].isbn = req.body.isbn
    autor[index].id_autor = req.body.id_autor

    res.status(200).json(autorLivro[index])
})

app.delete("/autorLivro/:isbn", (req, res) => {
    const index = buscarAutor(req.params.isbn)
    autorLivro.splice(index, 1)
    res.status(200).json(autorLivro)
});

////////////////////////////////////////////////////////////////////////
////////////////////////////////////////

function buscarExemplar(id_exemplar) {
    return exemplar.findIndex(exemplar => {
        return exemplar.id_exemplar === Number(id_exemplar)
    })
}

app.get("/", (req, res) => {
    res.status(200).send("Livraria Saber e Cia")
})

app.get("/exemplar", (req, res) => {
    res.status(200).json(exemplar)
})

app.get("/exemplar/:id_exemplar", (req, res) => {
    const index = buscarExemplar(req.params.id_exemplar)
    res.status(200).json(exemplar[index])
})

app.post("/exemplar", (req, res) => {
    exemplar.push(req.body)
    res.status(201).json(req.body)
})

app.put("/exemplar/:id_exemplar", (req, res) => {
```

```
    const index = buscarExemplar(req.params.id_exemplar)

    autor[index].id_exemplar = req.body.id_exemplar
    autor[index].status_exemplar = req.body.status_exemplar
    autor[index].isbn = req.body.isbn

    res.status(200).json(exemplar[index])
})

app.delete("/exemplar/:id_exemplar", (req, res) => {
    const index = buscarExemplar(req.params.id_exemplar)
    exemplar.splice(index, 1)
    res.status(200).json(exemplar)
});

////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////

function buscarMembro(id_membro) {
    return membro.findIndex(membro => {
        return membro.id_membro === Number(id_membro)
    })
}

app.get("/", (req, res) => {
    res.status(200).send("Livraria Saber e Cia")
})

app.get("/membro", (req, res) => {
    res.status(200).json(membro)
})

app.get("/membro/:id_membro", (req, res) => {
    const index = buscarMembro(req.params.id_membro)
    res.status(200).json(membro[index])
})

app.post("/membro", (req, res) => {
    membro.push(req.body)
    res.status(201).json(req.body)
})

app.put("/membro/:id_membro", (req, res) => {
```

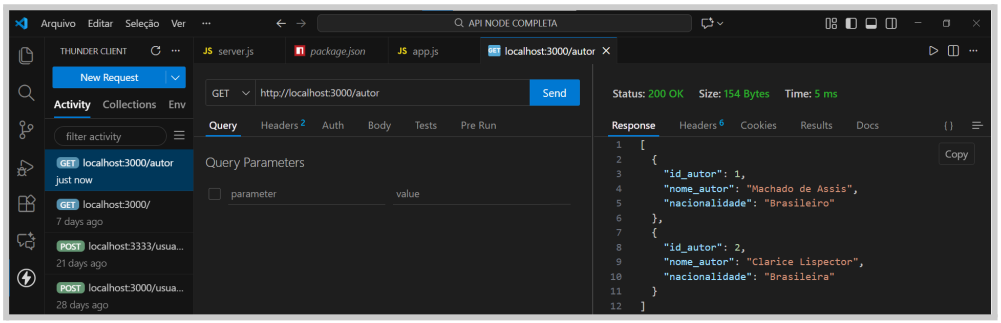
```
    const index = buscarMembro(req.params.id_membro)

    membro[index].id_membro = req.body.id_membro
    membro[index].nome_membro = req.body.nome_membro
    membro[index].endereco = req.body.endereco
    membro[index].telefone = req.body.telefone

    res.status(200).json(membro[index])
})

app.delete("/membro/:id_membro", (req, res) => {
    const index = buscarMembro(req.params.id_membro)
    membro.splice(index, 1)
    res.status(200).json(membro)
});

//////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////

function buscarEmprestimo(id_emprestimo) {
    return emprestimo.findIndex(emprestimo => {
        return emprestimo.id_emprestimo === Number(id_emprestimo)
    })
}

app.get("/", (req, res) => {
    res.status(200).send("Livraria Saber e Cia")
})

app.get("/emprestimo", (req, res) => {
    res.status(200).json(emprestimo)
})

app.get("/emprestimo/:id_emprestimo", (req, res) => {
    const index = buscarEmprestimo(req.params.id_emprestimo)
    res.status(200).json(emprestimo[index])
})

app.post("/emprestimo", (req, res) => {
    emprestimo.push(req.body)
    res.status(201).json(req.body)
})
```

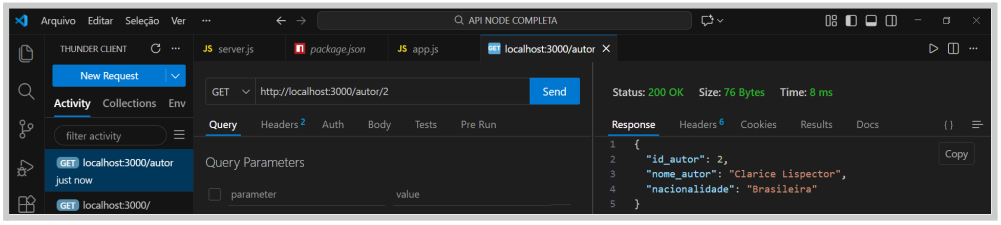
```
app.put("/emprestimo/:id_emprestimo", (req, res) => {
    const index = buscarEmprestimo(req.params.id_emprestimo)

    emprestimo[index].id_emprestimo = req.body.id_emprestimo
    emprestimo[index].data_emprestimo = req.body.data_emprestimo
    emprestimo[index].data_devolucao = req.body.data_devolucao
    emprestimo[index].data_devolucao_efetiva =
req.body.data_devolucao_efetiva
    emprestimo[index].id_exemplar = req.body.id_exemplar
    emprestimo[index].id_membro = req.body.id_membro

    res.status(200).json(emprestimo[index])
})

app.delete("/emprestimo/:id_emprestimo", (req, res) => {
    const index = buscarEmprestimo(req.params.id_emprestimo)
    emprestimo.splice(index, 1)
    res.status(200).json(emprestimo)
});

////////////////////////////////////////////////////////////////////////
////////////////////////////////////////

export default app
```

---

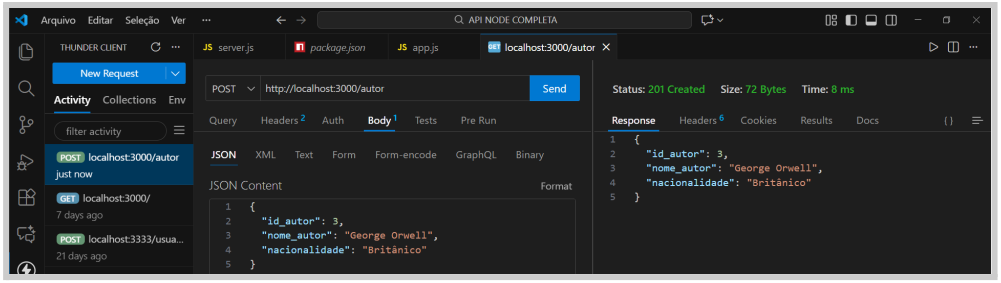
## Demonstração - Tabela AUTOR com comandos HTTP

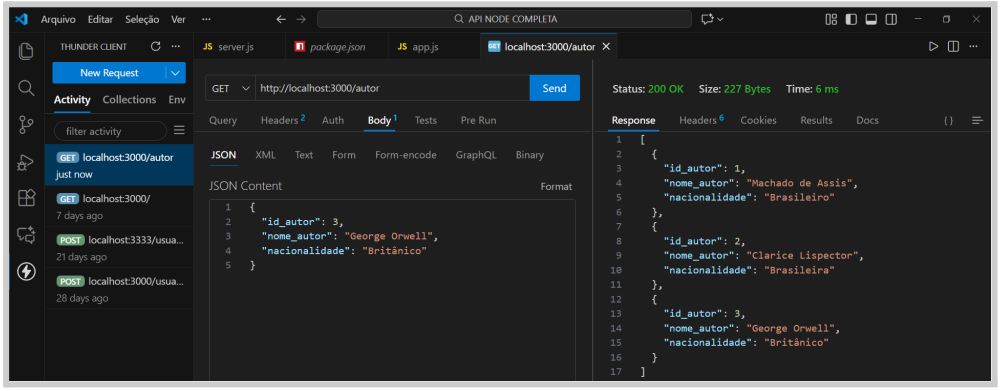
- Método **GET**:



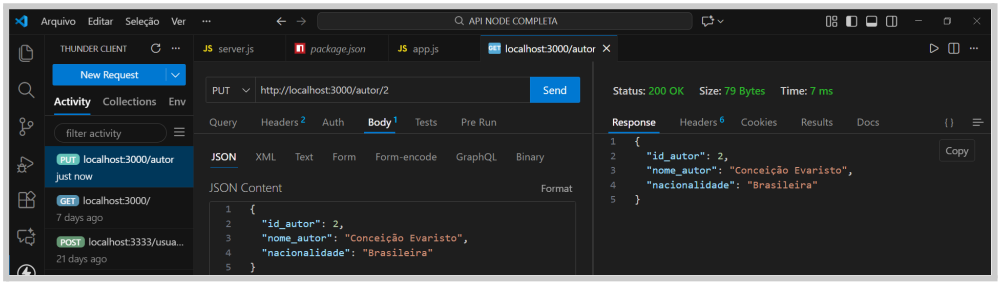



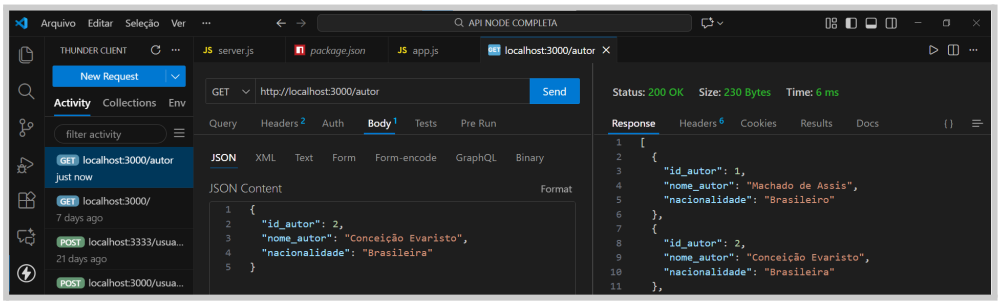

- Método **POST**:



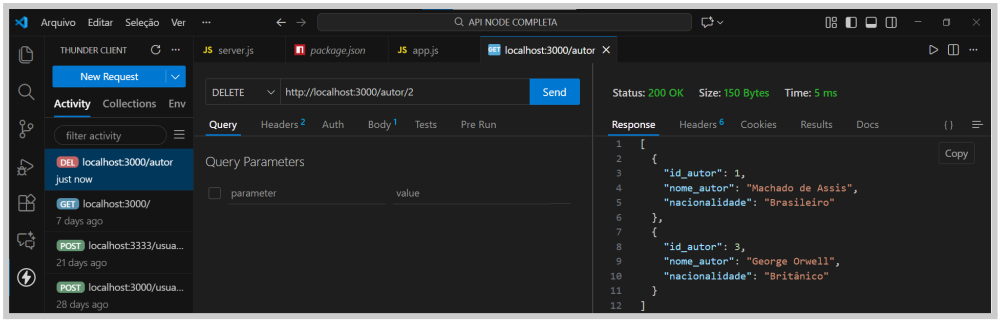



- Método **PUT**:




- Método **DELETE**:




---

## Breve Relatório Explicativo

Durante a criação da API, foram utilizados métodos HTTP, que são comandos padronizados que indicam a ação que um cliente (como navegador e no caso, o *Thunder Client*) deseja que o servidor execute um recurso específico. Em todas as entidades, foram testadas, separadamente, os métodos: GET, POST, PUT e DELETE.

O método **GET** é usado para solicitar um recurso do servidor padrão em navegadores, ele é usado para exibir uma lista (*array*) ou uma porta de um cômodo específico que podemos acessar para manipular algum recurso (*endpoint* ou *path parameters*), solicitando a representação de um recurso. O método **POST** é utilizado para enviar dados para o servidor para que eles os processe,como a criação de um

novo recurso. O método **PUT** é usado para atualizar um recurso existente por completo e, se o recurso não existir, ele pode ser criado. Já o método **DELETE**, remove um recurso específico do servidor.

As ferramentas utilizadas para a produção da API foi a biblioteca **Express**, que serve para criar servidores web e APIs usando o *Node.js*, simplificando o desenvolvimento ao fornecer ferramentas para gerenciar rotas, lidar com requisições HTTP e processar requisições e respostas.

A forma que usei para me organizar a estrutura do projeto foi a lógica do código. Primeiro, fiz a criação das entidades (tabelas ou *arrays*), adicionando seus respectivos atributos (como no caso do “livro”:  isbn, título do livro, editora e ano de publicação) e logo após implementei os métodos, na mesma ordem de antes.

As dificuldades que encontrei efetuando esse projeto foram: falta de atenção a escrita do código (alguns erros foram causados por erros de digitação ou a falta de alguma linha), os arquivos corrompidos na pasta *node_modules* da API e, principalmente, um erro ao rodar o servidor (no começo, não rodei pelo terminal, fazendo com que todos os arquivos “linkados” ao servidor não executarem corretamente).