# Creating a Node.js Server

**This content was extracted from Operações de CRUD em arquivo JSON usando JavaScript.*

The basic steps to create this Node.js server are:

```
> npm init -y
```

```
> npm install --save express
```

```
> npm install --save nodemon
```

You need to change some values of package.json file. These values are:

- **name**: you can use your ID/name. In my case, I used *ldmfabio*
- In *scripts*, the value of "dev" needs to be **"nodemon"**

In the package-lock.json file, all the value of *name* was changed to **"ldmfabio"**.

> The port number were the *back end* will be availeble was established at the last line of index.js file

Now you need to pay attention to the index.js file. In this file are all the HTML methods to alter the JSON file, where contains the data of the server.

To run the server, you need to use the command:

```
> npm run dev
```

For CRUD operations, you may use some tools like Postman or RapidAPI Client (my preferred one).

## RapidAPI

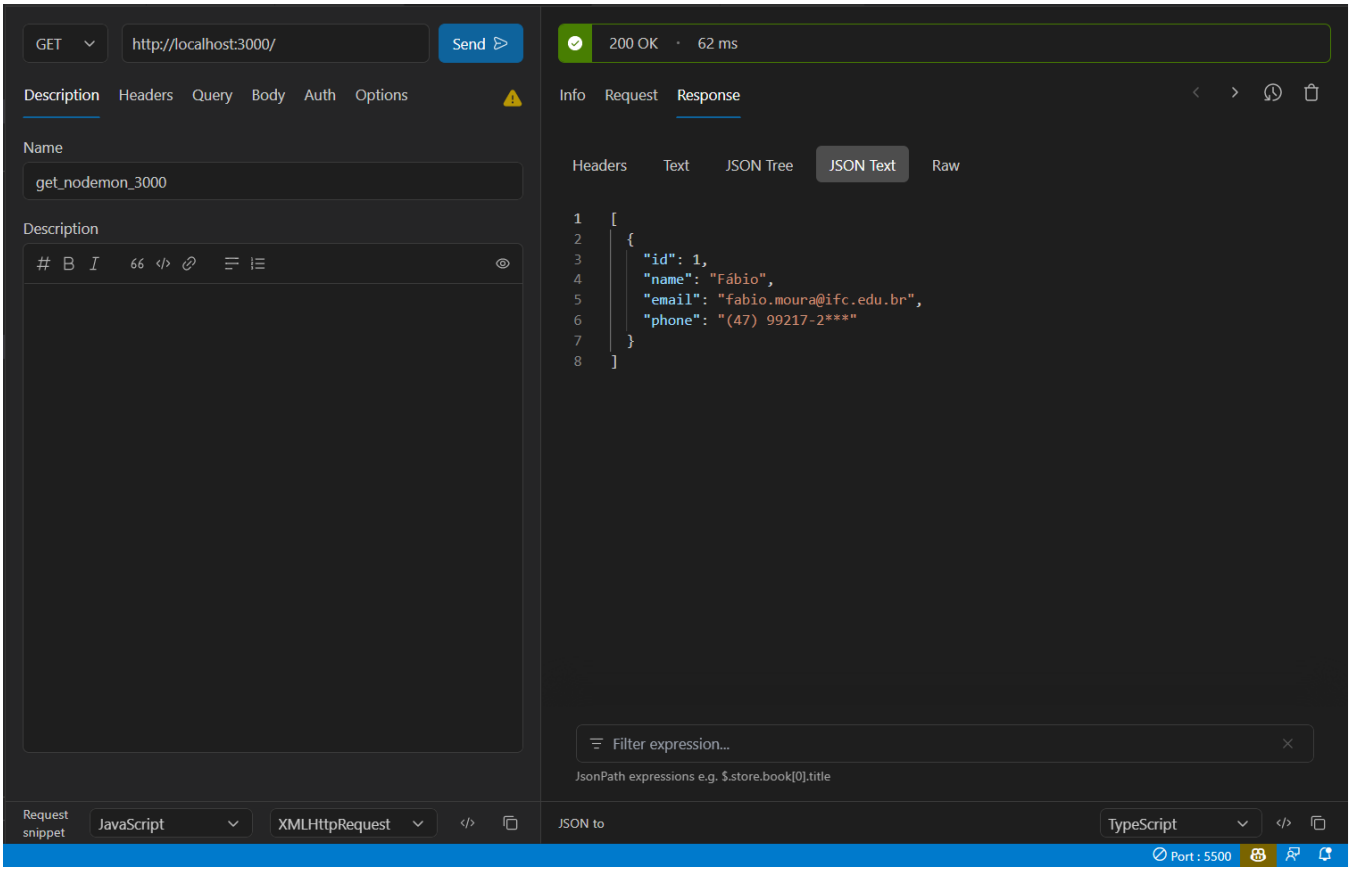
The JSON file (items.json) at the beginner is:

```
[
    {
        "id": 1,
        "name": "Fábio",
        "email": "fabio.moura@ifc.edu.br",
        "phone": "(47) 99217-2***"
    }
]
```

After *npm run dev* command, you can use the RapidAPI Client to do this is simple. You need to create a new request, set the method (GET, POST, PUT, DELETE), the URL (http://localhost:3000/ - after *npm run dev* being executed) and the body (if necessary).

## Get Method

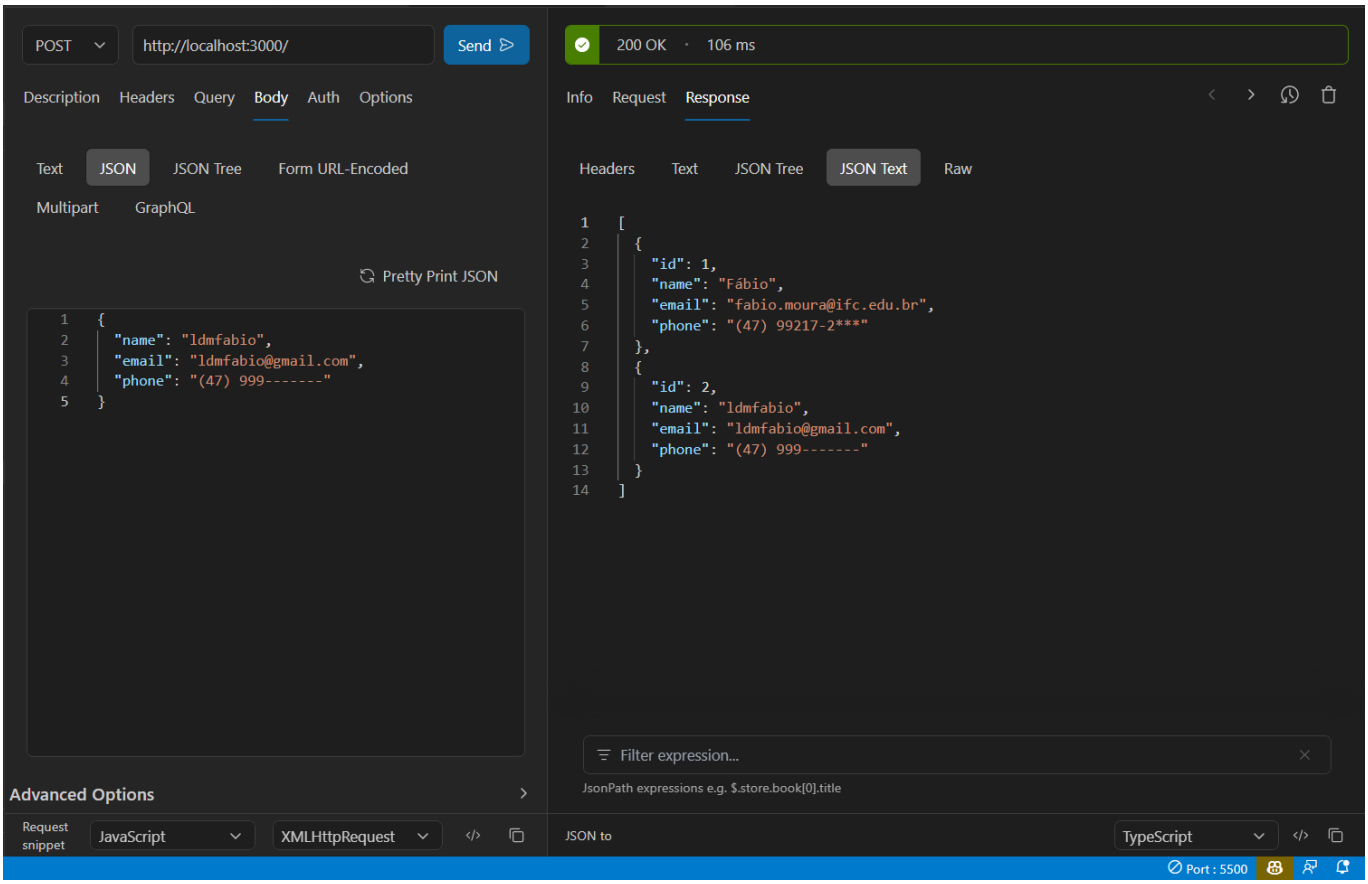
See the image below to see how to use the GET method and the result of this method.




*Click at the **Send** button to see the result of the GET method.*

## Post Method

See the image below to see how to use the POST method and the result of this method.

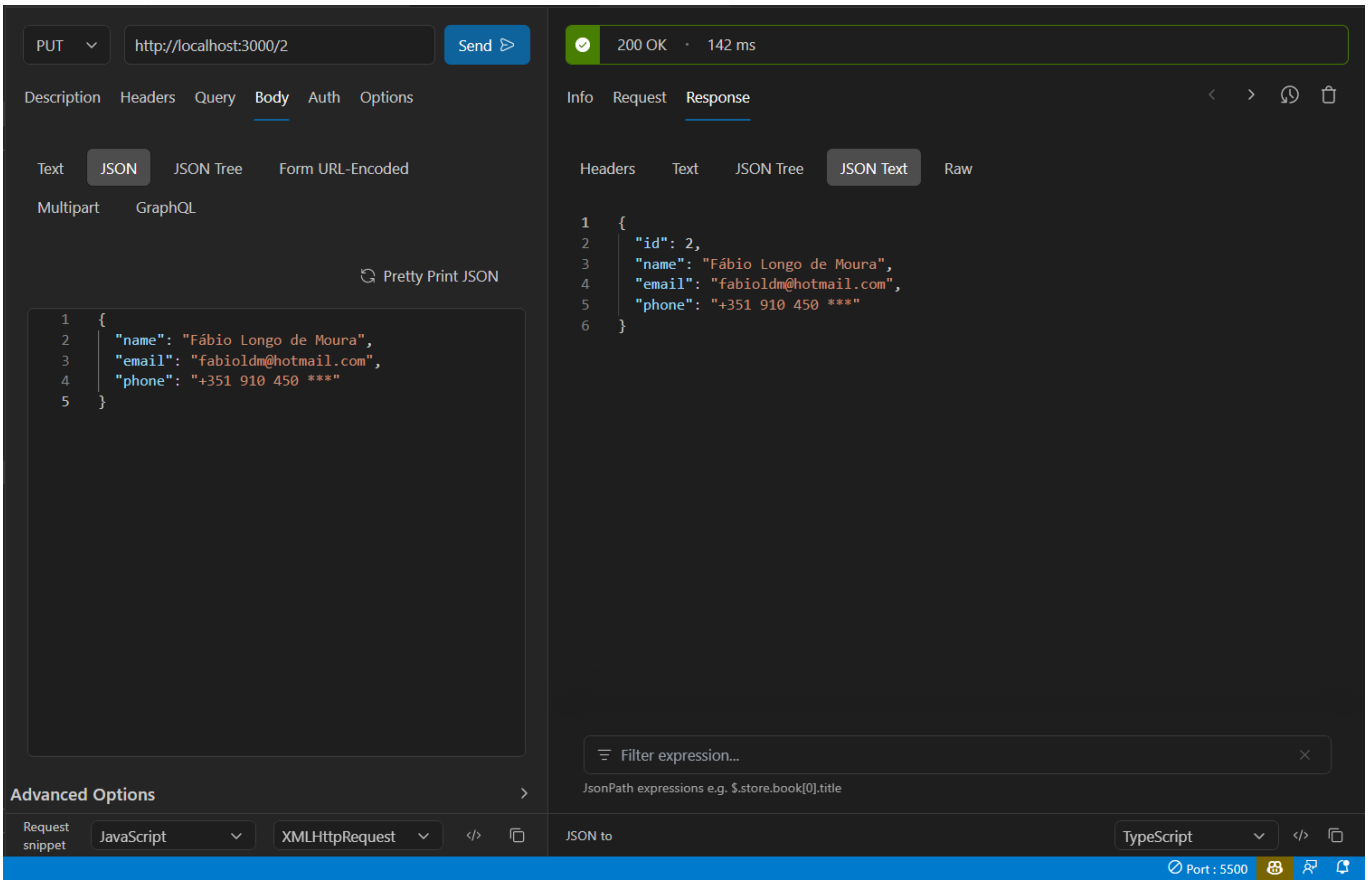



*Click at the **Send** button to see the result of the POST method.*

As you can see in the right side of the image, the JSON file was updated with the new data. Now, we have two records in the JSON file.

## Put Method

See the image below to see how to use the PUT method and the result of this method.

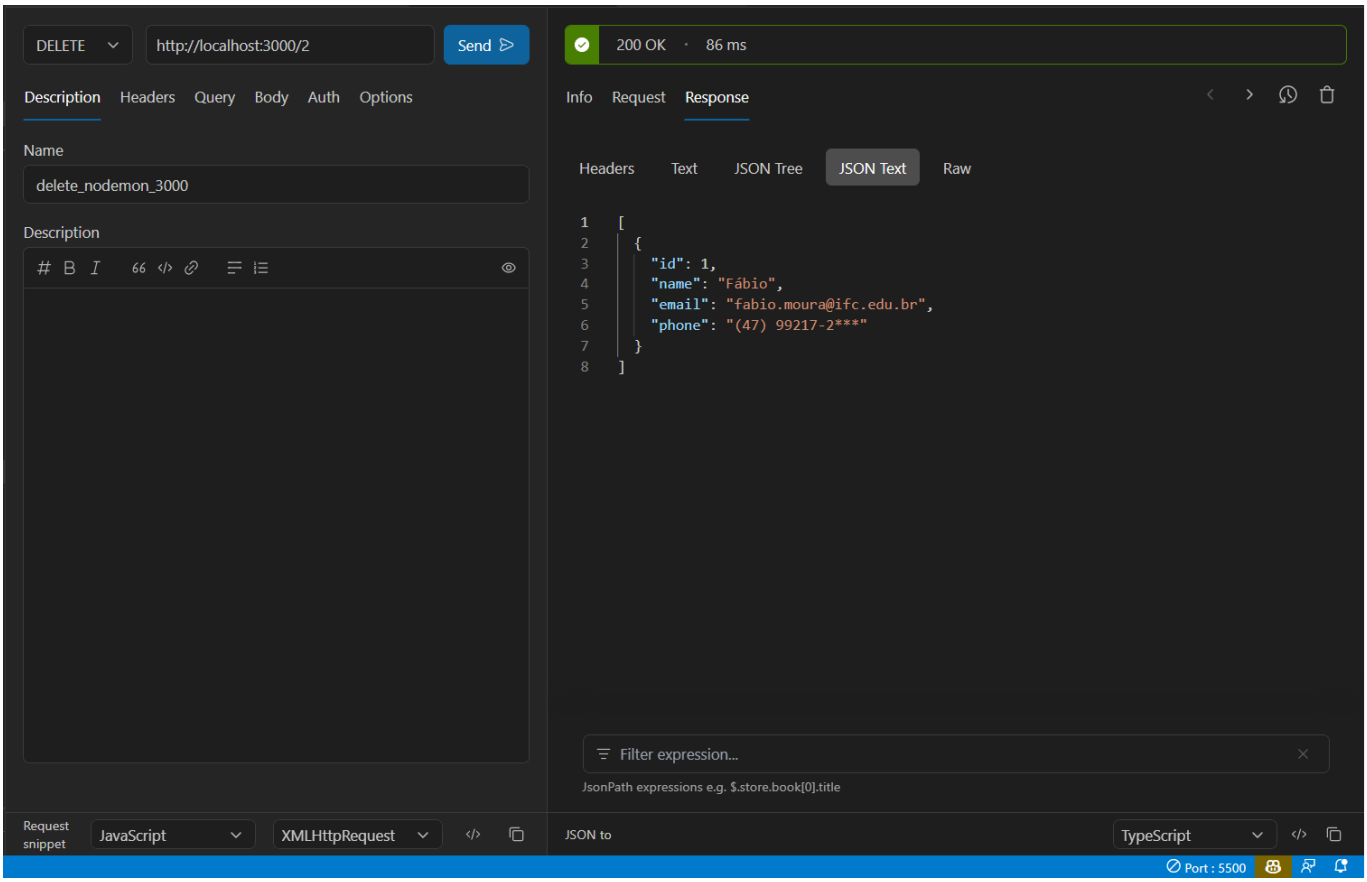



*Click at the **Send** button to see the result of the PUT method.*

As you can see in the right side of the image, the JSON file was updated with the new data displayed on the left side of the image. Beyond that, the PUT method is used to update the data of a specific record. In this case, the record with the ID 2 was updated. This parameter was set in the URL of the PUT method.

## Delete Method

See the image below to see how to use the DELETE method and the result of this method.




*Click at the **Send** button to see the result of the DELETE method.*

As you can see in the right side of the image, the JSON file was updated, but now, the record with the ID 2 was deleted. This ID was used in the DELETE method as a parameter.

# Making Swagger Documentation

**This content was extracted from Como automatizar documentação de APIs REST em Node.js com Swagger?.*

You first need to instal some npm dependencies. These dependencies are:

**swagger-ui-express, swagger-autogen and body-parser**

So, to install these dependencies, you need to use the command:

```
npm install --save swagger-ui-express swagger-autogen body-parser
```

Please verify if the dependencies were installed in the package.json file.

Now we need to create a new file and a new folder. The new file and folder are:

- **FILE:** swagger.js (at root folder)
- **FOLDER:** folder swagger (at root folder)

> *Both will be created at root folder*

## swagger.js

In this file, the code will be like this:

```
const swaggerAutogen = require('swagger-autogen')();

const outputFile = 'swagger/swagger_output.json';

const endPointsFiles = ['index.js'];

swaggerAutogen(outputFile, endPointsFiles);
```

Note that the swagger.js file will be used to generate the swagger_output.json file. This file will be used to generate the Swagger documentation.

The swagger documentation will be generated from the index.js file. This file contains all the methods to alter the JSON file.

## swagger folder

You don't need to create any file in this folder. The swagger-autogen will create all the files necessary to generate the Swagger documentation. This folder will just serve to store the files generated by swagger-autogen.

## index.js

At this file, you need to add some lines of needed code to generate the Swagger documentation. These lines are:

```
const bodyParser = require("body-parser");
const swaggerUi = require("swagger-ui-express");
const swaggerFile = require('./swagger/swagger_output.json');
```

> *Suggestion: add these lines after the line 1 of the index.js file.*

Also in the index.js file, you need to add some other lines of code to generate the Swagger documentation. These lines are:

```
server.use('/docs', swaggerUi.serve, swaggerUi.setup(swaggerFile));
server.use(express.json({ extended: true }));
server.use(bodyParser.urlencoded({ extended: false }));
```

> *Suggestion: add these lines before reading the JSON file (line 7 of the index.js file).*

## swagger docs

To generate the Swagger documentation, we need to change the package.json file. In this file, we need to go to section scripts to add swagger-autogen, according to the code below:

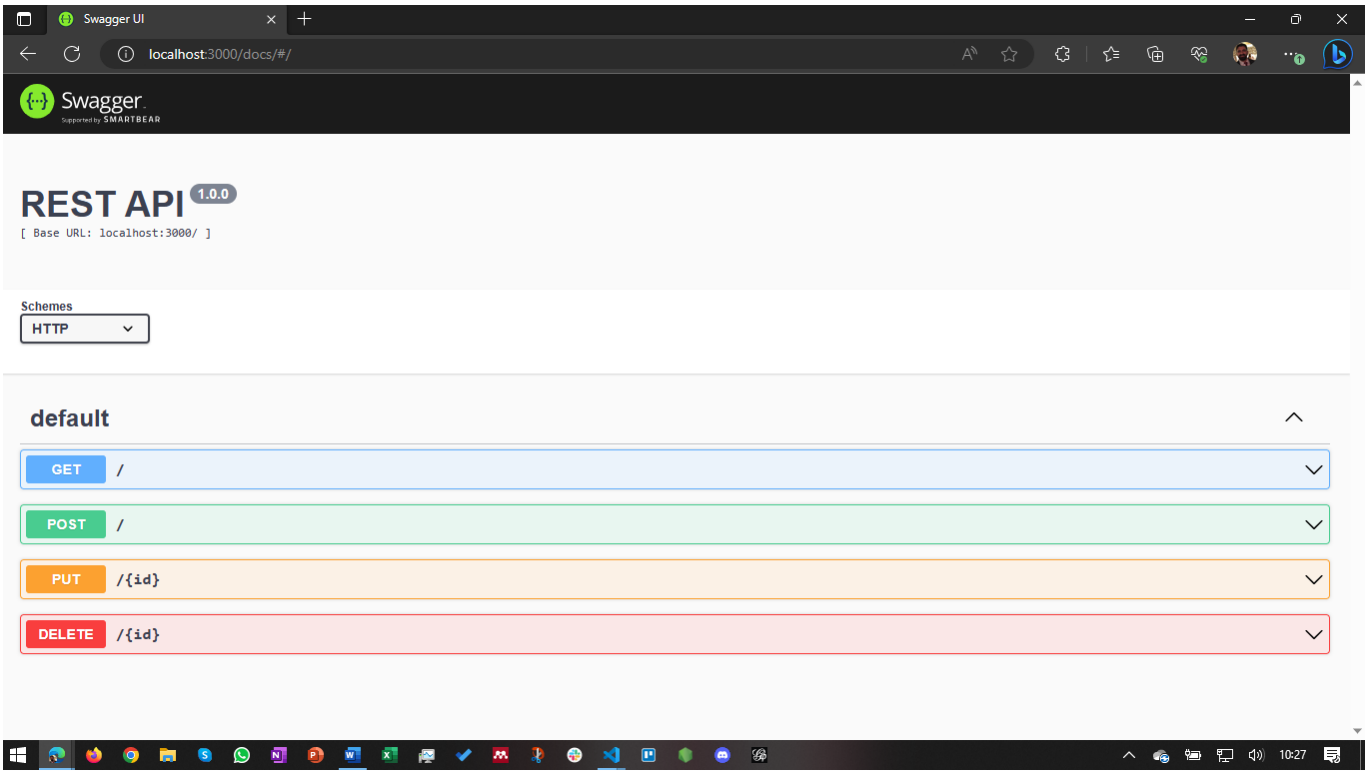
```
"scripts": {
  "start": "nodemon index.js",
  "swagger-autogen": "node swagger.js",
},
```

Now, to generate the Swagger documentation, you need to use the command:

```
npm run swagger-autogen
```

> *Pay attention at the swagger folder. Now you will se the file swagger_output.json. This file was generated based on the index.js file and their methods.*

To see the documentation, you just need to run the server and access the URL http://localhost:3000/docs. The result will be like this:




# References


- Operações de CRUD em arquivo JSON usando JavaScript.
- Como automatizar documentação de APIs REST em Node.js com Swagger?


## If you have any questions, please contact me.


- e-Mail: ldmfabio@gmail.com | fabio.moura@ifc.edu.br

- GitHub: github.com/ldmfabio
- LinkedIn: linkedin.com/in/ldmfabio
- WhatsApp: +55 (47) 99217-2425