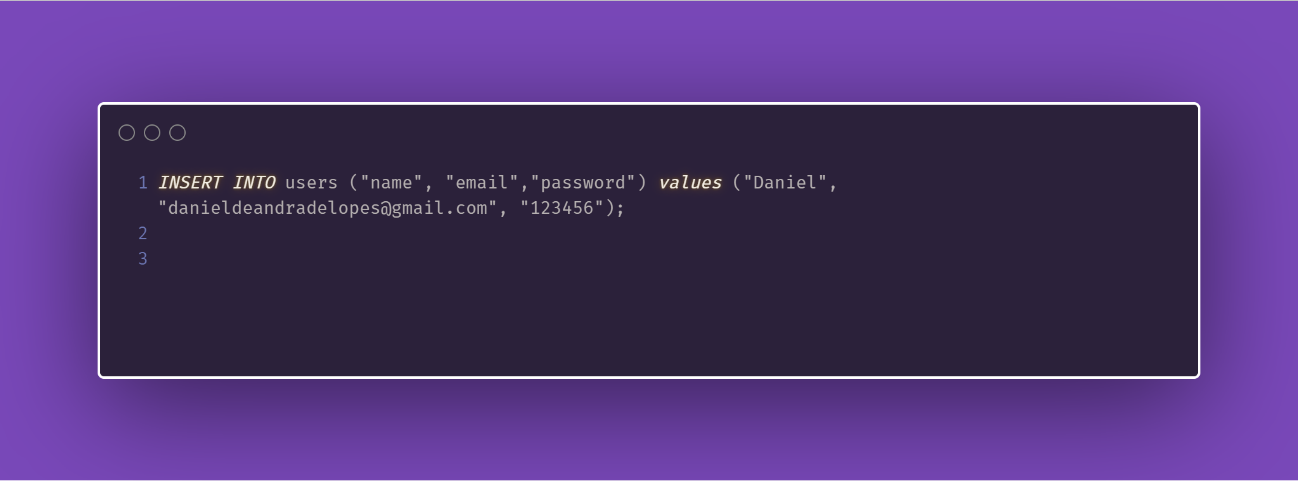
1. **O que é NodeJS?**
   1. É uma plataforma para construir aplicações web escaláveis de alta performance usando Javascript.
   2. Facilita a escrita de códigos SQL, pois, conhecendo Javascript você irá conseguir desenvolver um backend tranquilamente.

Veja abaixo como é um código SQL de inserção:



Já aqui temos o mesmo código utilizando Javascript (NodeJS):



Além de termos um código no backend utilizando o Javascript, podemos contar com os milhares de pacotes que o npm nos proporciona, tais como:

* Pacotes para manipulação de datas;
* Pacotes para upload de arquivos;
* Validações e muito mais.
  1. **Quem usa NodeJS?**
     1. LinkedIn
     2. Netflix
     3. IBM
     4. Uber
     5. Walmart
     6. Paypal
  2. **Produtividade:**
     1. Como a comunidade de Javascript é gigantesca, cada vez mais nós iremos ter pacotes, documentação e devs que colaboram tirando dúvidas e criando conteúdo, o que faz com que o NodeJS seja cada vez mais popular para os desenvolvedores.

1. **API Rest**
   1. Nas APIs (Application Programming Interface) funciona basicamente da seguinte maneira: Requisições x Respostas. Digamos que um cliente da API (quem está precisando dos dados) solicite ao servidor informações sobre os dados de um produto que está cadastrado no sistema, o servidor irá receber essa solicitação, irá processar os dados e irá “dar” um retorno para esse cliente, assim o cliente poderá processar o resultado obtido, o retorno das APIs são em formato JSON, veja baixo como ficaria os dados obtidos pelo cliente:



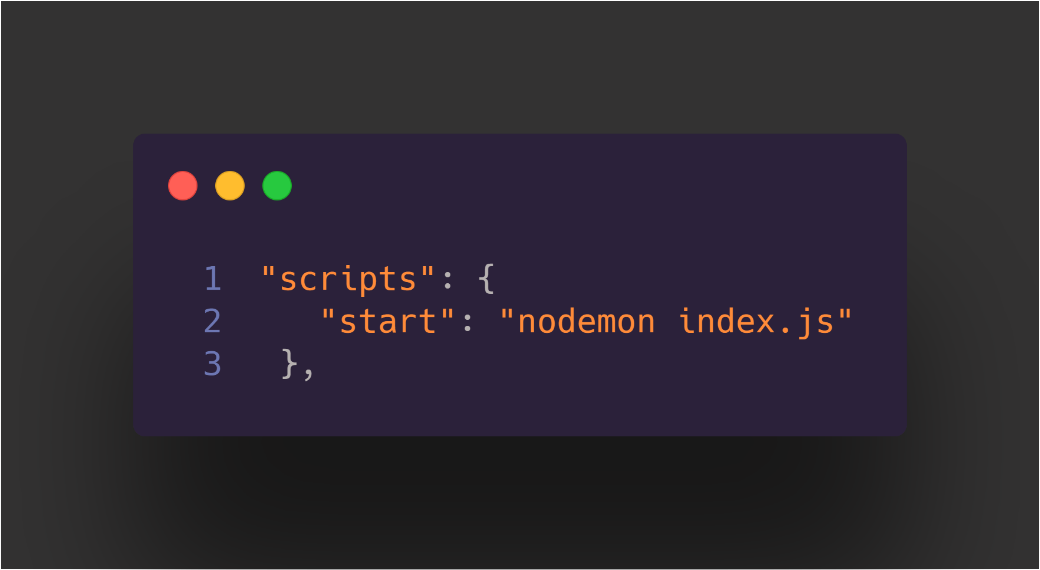
Dessa maneira o nosso front-end poderia “capturar” essas informações e exibir esses dados.

* 1. Uma das principais vantagens de se utilizar esse conceito de API Rest é que qualquer sistema poderá receber essas informações, seja ele Web, Mobile e até mesmo desktop, fazendo assim com que esse conceito seja muito útil, prático e econômico para a obtenção dos registros vindos de um banco de dados.

1. **Primeira aplicação com NodeJS**
   1. **Nessa aula nós iremos criar uma primeira aplicação utilizando NodeJS, para isso faremos o seguinte:**
      1. yarn init -y
      2. yarn add express
      3. criamos um arquivo index.js na raiz
      4. Deixamos o index.js assim:



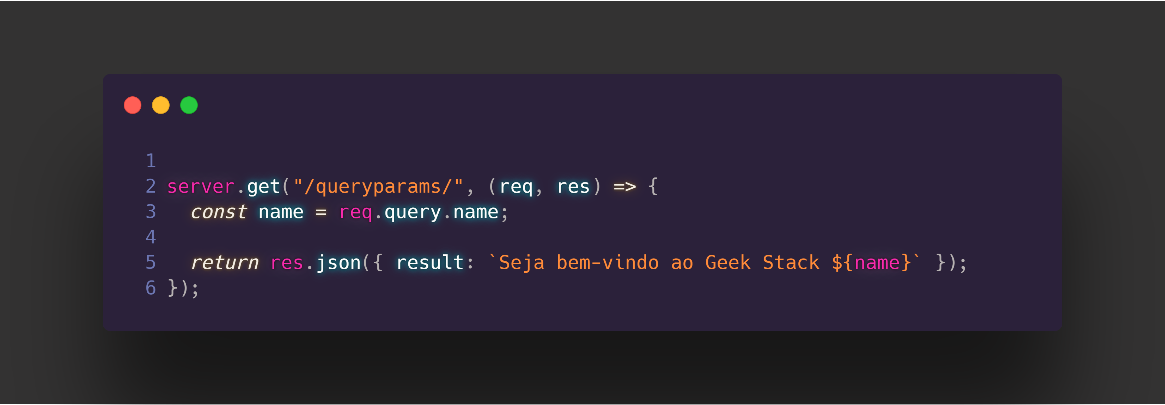
* + 1. Executamos o projeto com o comando: *node index.js*
    2. Para facilitar iremos instalar o nodemon usando o comando: yarn add nodemon -D (-d por conta de que iremos utilizar apenas em ambiente de desenvolvimento).
    3. Adicionamos scripts start no Package.json



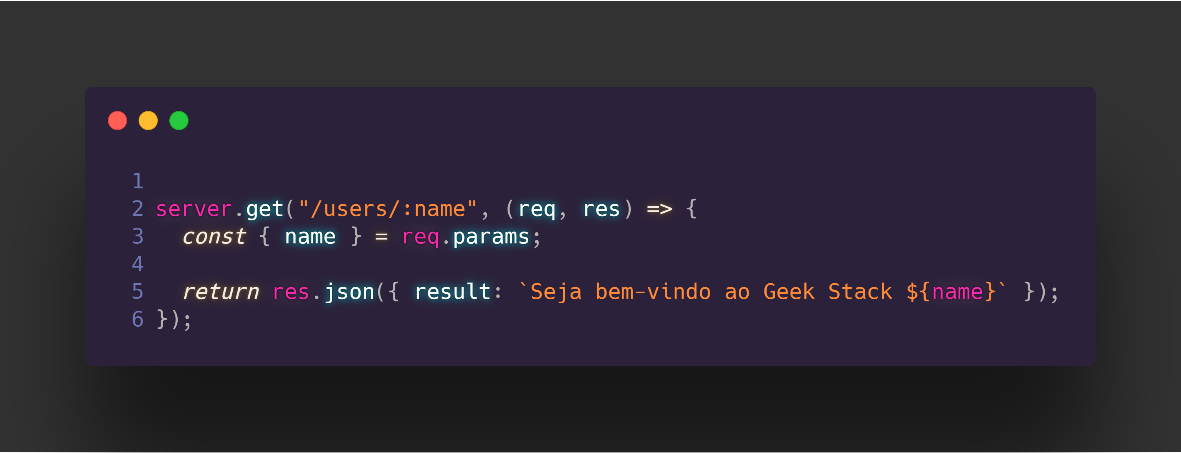
* + 1. Criação da primeira rota com NodeJS:



1. **Criando rotas com parâmetros:**
   1. Criamos uma segunda rota (Utilizando query params), os query params são aqueles passados na url da seguinte maneira: http://localhost:3000/**route\_name**/**?**name\_parameter=informacão



* 1. Criamos a segunda rota (utilizando Route params):



1. **Nessa aula iremos utilizar o Postman para poder enviar corpo para nossa requisição:**
   1. Primeiramente abrimos o nosso postman e iremos criar uma nova coleção.
   2. Recriamos as duas rotas (first e users) para mostrar a funcionalidade
   3. Criamos mais duas rotas para “cadastrar” um produto:



1. **Nessa aula nós iremos consumir uma API externa utilizando o AXIOS**
   1. yarn add axios (explicar que é para consumir APIs externas)
   2. acessar o site <https://swapi.co/> e mostrar o retorno da API pública de StarWars
   3. criar um arquivo api.js e preencher com as seguintes informações:



* 1. criar a rota para exibir as informações dos planetas:

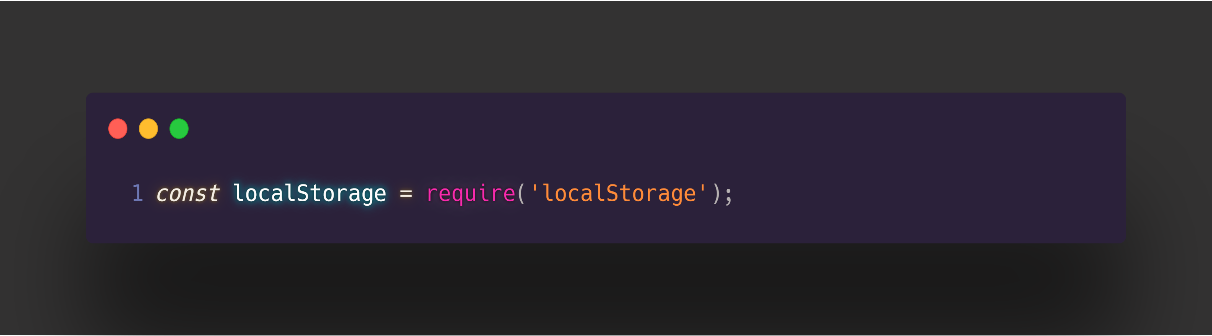


Depois nós iremos até o postman para fazer a requisição dessa rota.

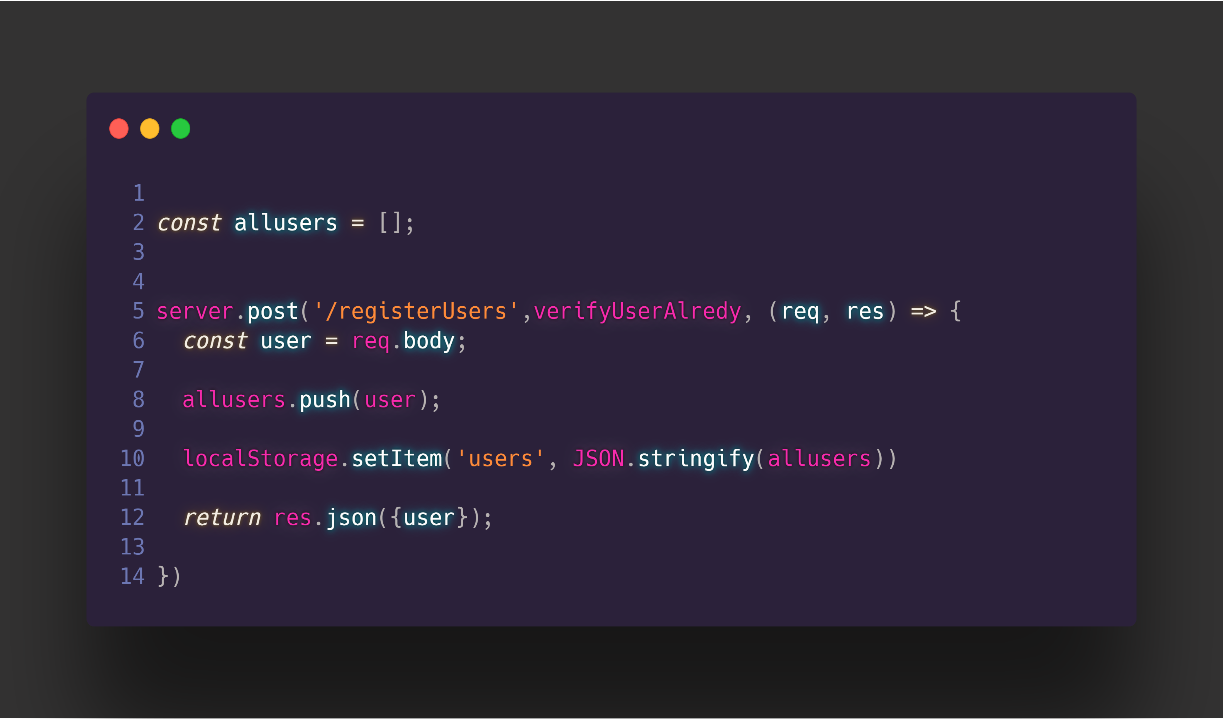
1. **Agora começaremos a trabalhar com Middlewares**

Middlewares tem uma importante função dentro do NodeJS, eles funcionam como uma placa de PARE, onde a nossa requisição irá “parar, olhar para um lado, olhar para o outro e então decidir se segue ou não”, basicamente os middlewares são uma função que verificam condições pré-definidas por nós, essas condições sendo satisfeitas as requisições continuam, caso contrário podemos realizar “outras” tarefas. Fique tranquilo, nós iremos ver isso na prática.

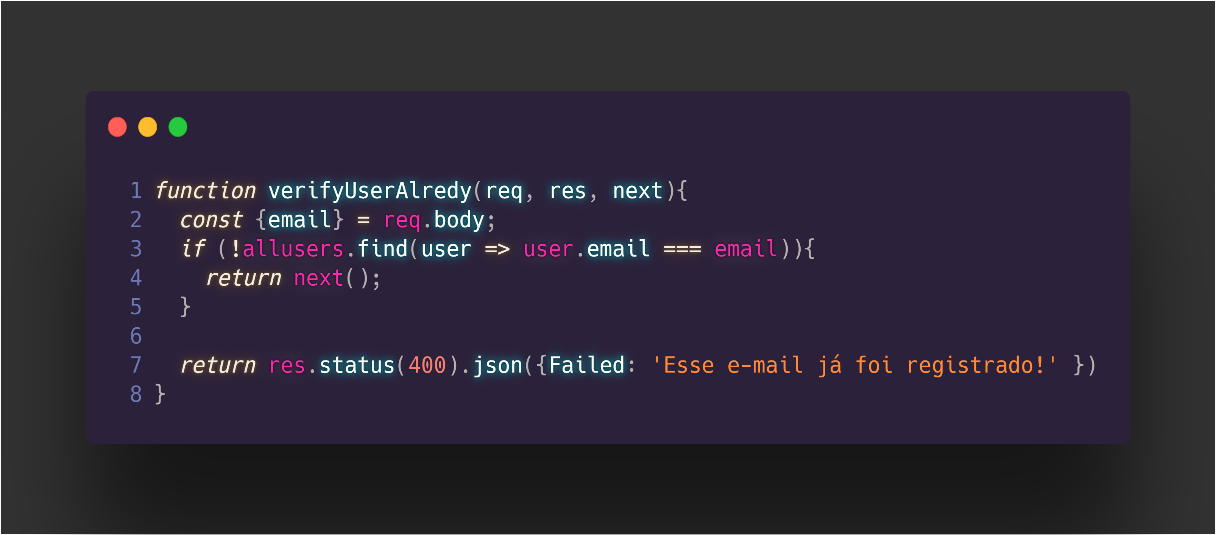
* 1. Começamos adicionando um pacote que permite que nós criemos um “banco de dados local”, visto que nessa aula nós iremos salvar as informações cadastradas lá pelo nosso **Postman**, esse pacote chama-se localstorage, para isso digite o seguinte comando: **yarn add localstorage**
  2. Faremos o import do localstorage para podermos trabalhar com ele:



* 1. Criamos um array e uma nova rota para popularmos (adicionar informações) ao nosso array:



* 1. Agora criamos o nosso middleware, veja que a sintaxe de um middleware é bem simples, é nada mais que uma função com parâmetros, note que essa função verifica se o e-mail que está sendo cadastrado não existe já no nosso “banco de dados”:



O middleware verifyUserAlready é um middleware específico para a rota (“/registerUsers”), mais a frente nós veremos middlewares que são GLOBAIS, ou seja, são aplicados a todas as rotas da nossa API.

* 1. Nesse passo basta ir até o **Postman** e fazer a requisição dessa rota (“/registerUsers”) e passar no body um JSON com as informações solicitadas no tópico “C”.
  2. No próximo passo nós iremos criar uma rota para exibir todos os nossos usuários cadastrados no sistema, o código ficará assim:



* 1. Para finalizar crie essa requisição no **Postman** e faça o teste.