

Criando um model de produtos

BackEnd

Vamos trabalhar agora com nossa API.

Criaremos o model de produtos.

Os produtos serão listados quando o cliente estiver logado.

O nome do model será produtoModel.js e será criado dentro da pasta model.

```
const Sequelize = require('sequelize');
const database = require('../db/db');

const Produto = database.define('produto', {
  id_produto: {
    type: Sequelize.INTEGER,
    autoIncrement: true,
    allowNull: false,
    primaryKey: true
  },
  nome: {
    type: Sequelize.STRING,
    allowNull: false,
  },
  descricao: {
    type: Sequelize.STRING,
    allowNull: false,
  },
  preco: {
    type: Sequelize.STRING,
    allowNull: false
  },
  link: {
    type: Sequelize.STRING,
    allowNull: false
  }
}, {database, modelName: 'produto', tableName: 'produtos'})
module.exports = Produto;
```

Agora vamos criar o controller do produto chamado produtoController.js dentro da pasta controller.

```
const Produto = require("../model/produtoModel");

module.exports = class produtoController{
  //CREATE
```

```
static async produtoCreate(req,res){
  let nome = req.body.nome;
  let descricao = req.body.descricao;
  let preco = req.body.preco;
  let link = req.body.link;

  const produto = {
    nome: nome,
    descricao: descricao,
    preco: preco,
    link: link
  }
  await Produto.create(produto);
  res.json({message: "Produto cadastrado com sucesso!"});
}

//READ - LISTAR
static async ProdutoListar(req,res){
  const id_produto = req.params.id;
  if(id_produto){
    const produto = await Produto.findOne({where: {id_produto: id_produto}});
    res.json(produto);
  }else{
    const produto = await Produto.findAll({raw:true});
    res.json(produto);
  }
}

//UPDATE
static async ProdutoUpdate(req, res){
  const id_produto = req.params.id;
  let nome = req.body.nome;
  let descricao = req.body.descricao;
  let preco = req.body.preco;
  let link = req.body.link;
  const produto = {
    nome: nome,
    descricao: descricao,
    preco: preco,
    link: link
  };
  await Produto.update(produto,{where: {id_produto:id_produto}});
  res.json({message: "Cadastro atualizado com sucesso! Foram atualizados as seguintes informações: ", dados: produto});
}

//Função ProdutoDelete responsável pela exclusão do usuário.
static async ProdutoDelete(req,res){
  const id_produto = req.params.id;
  await Produto.destroy({where:{id_produto: id_produto}});

  res.json({message: "Produto excluído com sucesso!"});
}
}
```

Agora vamos criar nossa view de cadastro de produtos e de listar produtos.

Crie uma pasta na nossa aplicação frontEnd dentro da pasta views com o nome de produtos.

Crie também um arquivo chamado Cadastrar.handlebars.

```
<h1 class="text-center my-3">Sistema de Cadastro de Produtos</h1>
<form action="/produtos/Cadastrar" method="post">
  <div class="row w-50 d-block m-auto g-3">
    <div class="col-12">
      <label class="form-label" for="nome">Nome do produto:</label>
      <input
        class="form-control"
        type="text"
        name="nome"
        id="nome"
        placeholder="Digite o nome do produto."
      />
    </div>
    <div class="col-12">
      <label class="form-label" for="descricao">Descrição do produto:</label>
      <input
        class="form-control"
        type="text"
        name="descricao"
        id="descricao"
        placeholder="Digite a descrição do produto."
      />
    </div>
    <div class="col-12">
      <label class="form-label" for="preco">Preço do produto:</label>
      <input
        class="form-control"
        type="text"
        name="preco"
        id="preco"
        placeholder="Digite o preço do produto." />
    </div>
    <div class="col-12">
      <label class="form-label" for="link">Link da imagem do produto:</label>
      <input
        class="form-control"
        type="text"
        name="link"
        id="link"
        placeholder="Digite o link da imagem do produto." />
    </div>
    <div>
      <input class="btn btn-primary my-3" type="submit" value="Cadastrar Produto" />
    </div>
  </div>
</form>
```

Vamos agora para o arquivo **services.js** para criar nossa função de envio para API.

```
//Create produto
static async ProdutoCreate(req,res){
  let valores = req.body;
  const options = {
    url: 'https://apivarejo.antonioizo.repl.co/add_produtos',
    method: 'POST',
    data: valores
  };
  axios(options);
  const mensagem = "Cadastro realizado com sucesso!";
  res.render("mensagem",{mensagem});
}
```

Vamos criar a rota de acesso ao formulário e a função de cadastro de produtos.

```
router.get("/produtos/Cadastrar",(req, res) =>{
  res.render("produtos/Cadastrar");
})
router.post("/produtos/Cadastrar",Services.ProdutoCreate);
```

Vamos na nossa API, no arquivo de routes.js, criaremos a rota para nossa função de create do controller.

```
//POST - CADASTRAR
router.post("/add_produtos", produtoController.produtoCreate);
```

Vamos aproveitar e criar a rota para listar produtos.

```
//GET - LISTAR
router.get("/produtos/:id?", produtoController.ProdutoListar);
```

Retornando para nossa aplicação, vamos criar agora a view para listar produtos.

Crie dentro da pasta produtos, um arquivo chamado listar.handlebars e cole o código abaixo.

```
<h1 class="text-center my-3">Lista de Produtos</h1>
<table class="table">
  <div class="row">
    <div class="col-12">
      <thead>
      </thead>
      <tbody>
        {{#each produto}}
          <tr>
            <th><center><br>
              <strong>{{this.nome}}</strong><br>
              {{this.descricao}} <br>
              R$ {{this.preco}} <br>
              cod. Produto: {{this.id_produto}}<br>
              <a href="/carrinho/Adicionar"></a>
            </th>
            <form class="borde-0 m-0" action="/carrinho/Adicionar" method="POST">
              <input type="hidden" name="id_produto" value="{{this.id_produto}}>
            </form>
          </tr>
        {{/each}}
      </tbody>
    </div>
  </div>
</table>
```

O código lista os produtos da tabela produtos.

Vamos no arquivo services.js para criar a função

```
//LISTAR
static async ProdutoListar(req,res){
  const options = {
    url: 'https://apivarejo.antonioizo.repl.co/produtos',
    method: 'GET',
    data: {}
  };
  axios(options).then(response => {
    console.log(response.data);
    const produto =response.data

    res.render("produtos/listar",{produto});
```

```
});  
}
```

No arquivo routes.js, vamos colocar a nossa rota.

```
router.get("/produtos/listar", Services.ProdutoListar);
```

Pronto agora só precisamos excluir o banco de dados da API e executar novamente para criar a tabela produtos.

Na próxima aula vamos testar nossa aplicação e verificar nossas rotas.

Até a próxima aula.