

Envio de imagens

Baixe a pasta Upload_Imagem353 no site www.evandro.jv.com.br → 354

Acesse a pasta e entre no Prompt (cmd)

Vamos fazer as configurações e instalar pacotes necessários

```
Upload_imagem_354>npm init -y
```

```
Upload_imagem_354>npm i express ejs sequelize body-parser mysql2 --save
```

RODE O SERVIDOR e vamos ver se aparece a seguinte tela abaixo

Envio de Imagens - 354 - Seu Nome

Nome Produto:

Valor Produto:

Escolher arquivo

Nenhum arquivo escolhido

Cadastrar

Abra o Xampp e crie o banco “empresa354”

Dentro da pasta já existe o a pasta “database” lembrando aqui fazemos as configurações do banco e tabela

Vamos criar dentro dela o arquivo “basedados.js” Atenção ali no lugar de empresa coloque o nome do seu banco

```
const Sequelize = require('sequelize');
const conexao = new Sequelize("empresa", "root", "", {
  host: "localhost",
  dialect: "mysql",
  define: {
    timestamps: false,
    freezeTableName: true,
  },
});
module.exports = conexao;
```

Depois vamos criar o arquivo “Produto.js” é o nosso “model” o responsável por cuidar da tabela que vamos criar no banco.

```
const Sequelize = require("sequelize");
const conexao = require("../basedados");

const Produto = conexao.define("produto", {
  id_produto: {
    type: Sequelize.INTEGER,
    autoIncrement: true,
    primaryKey: true,
  },
  nome: {
    type: Sequelize.STRING,
    allowNull: false,
  },
  preco: {
    type: Sequelize.REAL,
    allowNull: false,
  },
  imagem: {
    type: Sequelize.STRING,
    allowNull: false,
  },
});

Produto.sync();
module.exports = Produto;
```

Vamos restartar o servidor e ver se conectou com o banco e criou a tabela “produto”.

Vamo no nosso arquivo “index.js “ e fazer as configurações para acessar o nosso banco e tabela.

```
const express = require('express');
const app = express();
const bodyParser = require("body-parser");
const conexao = require('../database/basedados');

const Produto = require('../database/Produto');

conexao.authenticate().then(() => {
  console.log("CONECTADO COM O BANCO")
}).catch((errorMsg) => {
  console.log(errorMsg);
})
```

Restarte o servidor e verifique se a tabela “produto” foi devidamente criada

Vamos criar a a rota do “post” e vamos verificar se os dados estão chegando.

```
app.get("/", (req, res) => {
  res.render("enviar")
})

app.post("/cadastrar", (req, res) => {
  console.log(req.body);
  res.end();
})
```

Se tentarmos cadastrar podemos notar que nosso “console.log” está retornando vazio

```
SERVIDOR RODANDO
Executing (default): SELECT 1+1 AS result
Executing (default): SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_TYPE = 'BASE TABLE' AND TABLE_NAME = '
produto' AND TABLE_SCHEMA = 'empresa353'
CONECTADO COM O BANCO
Executing (default): SHOW INDEX FROM `produto`
{}
```

Isso porque temos um parâmetro chamado “files” e está conflitanto dentro do Node.js

Para isso vamos baixar o modulo de upload

```
Upload_imagem_353>npm i express-fileupload --save
```

E vamos configurar ele dentro do nosso “index.js”

```

const express = require('express');
const app = express();
const bodyParser = require("body-parser");
const conexao = require('./database/basedados');
const ControleProduto = require('./database/Produto');

const fileupload = require('express-fileupload');

conexao.authenticate().then(() => {
  console.log("CONECTADO COM O BANCO")
}).catch((errorMsg) => {
  console.log(errorMsg);
})

app.use(fileupload());

app.set("view engine", "ejs");
app.use(express.static("public"));
app.use(bodyParser.urlencoded({ extended: true }));

```

Agora tente enviar novamente

```

SERVIDOR RODANDO
Executing (default): SELECT 1+1 AS result
Executing (default): SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_TYPE = 'BASE TABLE' AND TAB
produto' AND TABLE_SCHEMA = 'empresa353'
CONECTADO COM O BANCO
Executing (default): SHOW INDEX FROM `produto`
[Object: null prototype] { nome: 'Camisa Baluda', valor: '256,20' }

```

Veja que chegou os dados de nome e preço e o nome da imagem, vamos arrumar a nossa rota "post"

```

app.get("/", (req, res) => {
  res.render("enviar")
})

app.post("/cadastrar", (req, res) => {
  console.log(req.body);
  console.log(req.files.imagem.name)
  res.end();
})

```

Vamos testar para ver se o nome da imagem aparece.

```

SERVIDOR RODANDO
Executing (default): SELECT 1+1 AS result
Executing (default): SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_TYPE =
produto' AND TABLE_SCHEMA = 'empresa353'
CONECTADO COM O BANCO
Executing (default): SHOW INDEX FROM `produto`
[Object: null prototype] { nome: 'Camisa Baluda', valor: '256,20' }
imagem1.png

```

Então vamos cadastrar no banco e fazer o upload da imagem para a devida pasta

```

app.get("/", (req, res) => {
  res.render("enviar")
})

app.post("/cadastrar", (req, res) => {
  let nome = req.body.nome;
  let preco = req.body.valor;
  let imagem = req.files.imagem.name;
  Produto.create({
    nome: nome,
    preco: preco,
    imagem: imagem
  }).then(() => {
    req.files.imagem.mv(__dirname + '/upload/' + req.files.imagem.name)
    res.redirect("/")
  })
})

```