



PRÁTICAS AVANÇADAS EM DESENVOLVIMENTO WEB

Davi Schneid - davi.Schneid@gmail.com

16/07/2024

Agenda

- ▶ Consumir APIs externas
- ▶ Utilizar o pacote Axios
 - ▶ Introdução
 - ▶ Visitando <https://axios-http.com/>

Instalar npm install axios

- ▶ Executar o comando no terminal: npm install axios

```
PS C:\Users\SenacRs\PraticasAvancadasDesenvolvimentoWeb\api-web> npm install axios
added 3 packages, and audited 1549 packages in 8s

261 packages are looking for funding
  run `npm fund` for details

8 vulnerabilities (2 moderate, 6 high)

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
```

- ▶ Para usar o cliente HTTP, preciso importar na página
 - ▶ import axios from 'axios';

API Web

► Editar o arquivo Cadastro.js

```
function Cadastro() {  
  useEffect(() => {  
    axios.get('https://servicodados.ibge.gov.br/api/v1/localidades/estados')  
      .then(response => {  
        console.log(response.data);  
      })  
  }, []);  
  
  return (  

```

Formulario de Cadastro

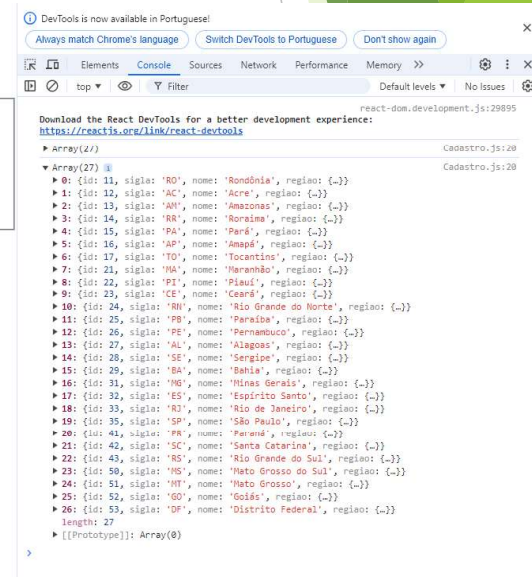
Dados de Cadastro

Nome:

Idade:

Cidade:

UF:



API Web

- ▶ Renderizar os valores carregados no campo
- ▶ Editar o arquivo Cadastro.js adicionando o código abaixo.

```
src > paginas > JS Cadastro.js > ...  
4 import '../App.css';  
5  
6 import BotaoVoltar from '../componentes/BotaoVoltar';  
7  
8 //Importar o cliente HTTP para requisicoes api externas  
9 import axios from 'axios';  
10  
11 //Utilizada para auxiliar no controle de outras funcoes da aplicacao  
12 import { useEffect, useState } from 'react';  
13
```

```
14  
15 function Cadastro() {  
16  
17 const [estados, setEstados] = useState([]);  
18  
19   useEffect(() => {  
20     axios.get('https://servicodados.ibge.gov.br/api/v1/localidades/estados')  
21       .then(response => {  
22         setEstados(response.data);  
23       })  
24   }, []);  
25  
26   return (  
27
```

```
    <div>  
      <label>UF:  
        <select name="cmbUF" id="cmbUF" >  
          <option value="0">Selecione uma opção</option>  
          {estados.map(estado => (<option key={estado.sigla} value={estado.sigla}>{estado.sigla}</option>))}  
        </select>  
      </label>  
    </div>
```

API Web

- ▶ Ao recarregar a página de cadastro.
- ▶ As UF serão apresentadas no combobox

Formulario de Cadastro

Dados de Cadastro

Nome:

Idade:

Cidade:

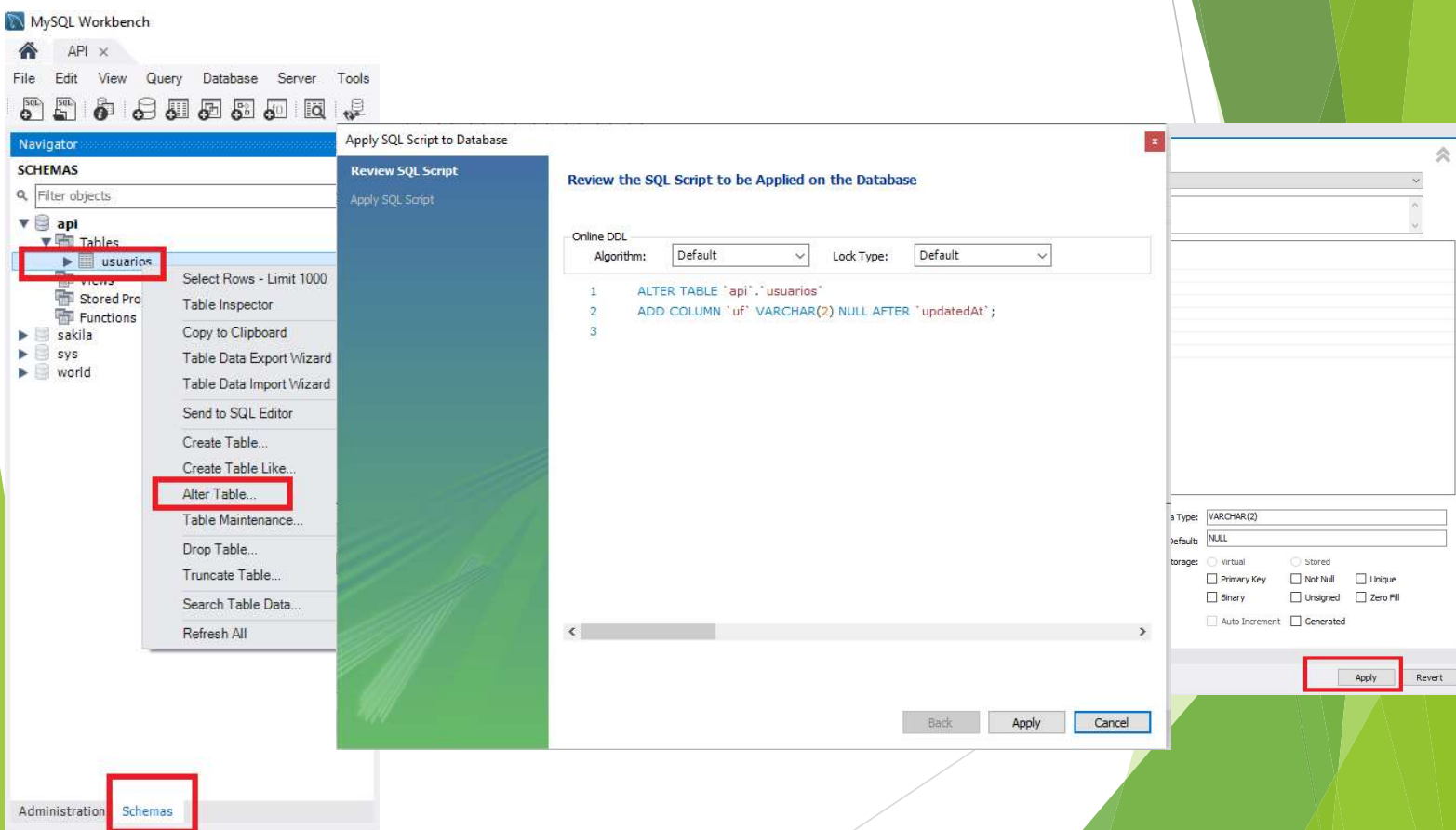
UF:

Selecione uma opção

RO
AC
AM
RR
PA
AP
TO
MA
PI
CE
RN
PB
PE
AL
SE
BA
MG
ES
RJ

API Router

- ▶ Abrir o Mysql Workbench
- ▶ Alterar a tabela usuários no banco de dados adicionando a coluna UF



API Router

- ▶ Alterar modelo Usuario.js incluindo novo campo UF.
- ▶ Alterar controller UsuarioController.js incluindo no campo UF

```
const Usuario = database.define('usuario', {  
  id: {  
    type: Sequelize.INTEGER,  
    autoIncrement: true,  
    allowNull: false,  
    primaryKey: true  
  },  
  nome: {  
    type: Sequelize.STRING,  
    allowNull: false  
  },  
  idade: {  
    type: Sequelize.INTEGER,  
    allowNull: false  
  },  
  cidade: {  
    type: Sequelize.STRING,  
    allowNull: false  
  },  
  uf: {  
    type: Sequelize.STRING,  
    allowNull: true  
  }  
}, {
```

```
8  
9 // Criar um novo usuário  
10 exports.createusuario = async (req, res) => {  
11   console.log('createusuario');  
12   const { nome, idade, cidade, uf } = req.body;  
13   console.log('Createusuario.Nome'+nome);  
14   console.log('createusuario.Idade'+idade);  
15   console.log('createusuario.Cidade'+cidade);  
16   console.log('createusuario.UF'+uf);  
17   try {  
18     const novoUsuario = await Usuario.create({ nome, idade, cidade, uf });  
19     res.status(201).json(novoUsuario);  
20   } catch (err) {  
21     console.log("Erro ao criar usuário");  
22     res.status(500).json({ error: 'Erro ao criar usuário' });  
23   }  
24 };
```

```
36 // Atualizar um usuário  
37 exports.updateusuario = async (req, res) => {  
38   const { id } = req.params;  
39   const { nome, idade, cidade, uf } = req.body;  
40   console.log('updateusuario id: ' + id + ' nome: ' + nome + ' idade: ' + idade + ' cidade: ' + cidade + ' uf: ' + uf);  
41   try {  
42     const usuario = await Usuario.findByPk(id);  
43     if (usuario) {  
44       usuario.nome = nome;  
45       usuario.idade = idade;  
46       usuario.cidade = cidade;  
47       usuario.uf = uf;  
48       usuario.updatedAt = new Date();  
49       await usuario.save();  
50       res.status(200).json(usuario);  
51     } else {  
52       res.status(404).json({ error: 'Usuário não encontrado' });  
53     }  
54   } catch (err) {  
55     res.status(500).json({ error: 'Erro ao atualizar usuário' });  
56   }  
57 };
```


API Web

- ▶ Editar o Cadastro.js adicionando useEffect no import

```
//Utilizada para auxiliar no controle de outras funcoes da aplicacao
import React, { useState, useEffect } from 'react';
```

- ▶ Adicionar o campo uf no useState setCampos

```
//cria novo estado para os campos da tela
const [campos, setCampos] = useState({
  nome: '',
  idade: 0,
  cidade: '',
  uf: ''
});
```

- ▶ Iniciar o useState do array de estados

```
const [estados, setEstados] = useState([]);

useEffect(() => {
  axios.get('https://servicodados.ibge.gov.br/api/v1/localidades/estados')
    .then(response => {
      setEstados(response.data);
    })
}, []);
```

- ▶ Adicionar a validação do campo UF na função validarCampos

API Web

► Exercício

- Atualizar a página EditarRegistro.js adicionando o campo UF, validação e etc.
- Atualizar a página ListaRegistro.js, adicionando o campo UF.