

Aula 6 - Requisições axios

Docupedia Export

Author:Ferro Alisson (CtP/ETS)

Date:01-Sep-2023 13:44

Table of Contents

1 Requisições GET	4
2 Requisições POST	6
3 Desafio 1: Faça um formulário de músicas com título, nome do autor, nome da gravadora, ano de lançamento e envie para a API https://jsonplaceholder.typicode.com/posts	7
4 Desafio 2: utilizando uma API pública crie um site para realizar as requisições necessária. Crie no mínimo duas página, uma geral, e uma para detalhar um item em específico.	8

Até o momento, nosso APP não fazia requisições, todas as informações ficavam na página somente e caso recarregasse perderia todos os dados, Mas primeiro, o que é uma requisição HTTP?

Uma requisição HTTP é um pedido feito por um cliente a um servidor para obter ou enviar informações na Web. O HTTP é o protocolo fundamental usado para comunicação entre navegadores da web e servidores web. Quando você acessa um site, clica em um link ou envia um formulário online, está fazendo requisições HTTP para obter páginas da web, recursos como imagens, vídeos e outros tipos de conteúdo, ou para interagir com aplicativos web.

O axios é uma biblioteca em que permite que façamos requisições para nosso *back-end* de forma simples, para isso basta instalar na nossa aplicação com o comando e esperar até que a instalação seja concluída

```
npm i axios
```

1 Requisições GET

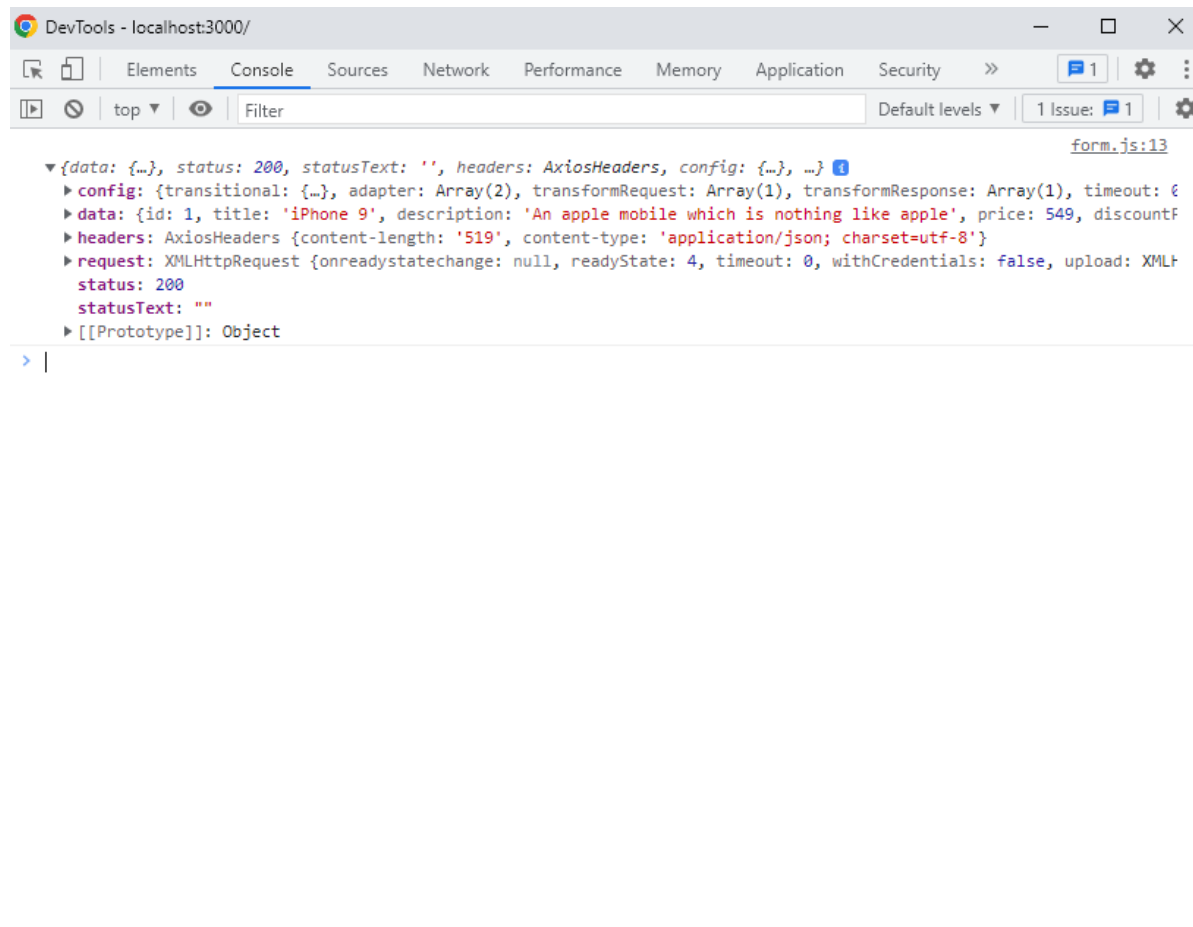
Após a instalação concluída, basta importarmos no componente que utilizará a requisição e chamar a função

```
import { useEffect } from 'react';
import axios from 'axios';

export default function ReqPage(){
  useEffect(() => {
    handleGet();
  }, []);
  async function handleGet(){
    const res = await axios.get('https://dummyjson.com/products/1');
    console.log(res);
  }

  return(
    <>Teste</>
  )
}
```

Observe que por se tratar de requisições para o *back-end* a função deve ser *async await* senão não poderemos imprimir nenhum resultado. Ao salvar o arquivo e terminada a compilação vamos abrir o console, uma mensagem no console deve aparecer logo após a requisição ser concluída



Onde se trata de uma resposta JSON, e podemos expandir para visualizar a mensagem completa, observe que em *data* estão os resultados em que precisamos. Agora basta setar as variáveis que vem do JSON. Uma boa pratica também é fazer um componente de error que será exibido caso ocorra algum erro.

2 Requisições POST

Para fazermos requisições do tipo POST, é um pouco semelhante as requisições GET, com a diferença que agora alteramos o método para post e também agora enviamos um body na requisição.

```
const handlePost = useCallback(async () => {
  const res = await axios.post('https://jsonplaceholder.typicode.com/posts', {
    title: "Criando com axios",
    content: "Algum conteúdo"
  });
  console.log(res);
}, [])

useEffect(() => {
  handlePost()
}, [handlePost])
```

Após salvar, vamos ver qual será a resposta no navegador.

```
▼ {data: {...}, status: 201, statusText: '', headers: AxiosHeaders, config: {...}, ...} ⓘ
  ► config: {transitional: {...}, adapter: 'xhr', transformRequest: Array(1), transformResponse: Array
  ▼ data:
    content: "Algum conteúdo"
    id: 101
    title: "Criando com axios"
    ► [[Prototype]]: Object
  ► headers: AxiosHeaders {cache-control: 'no-cache', content-length: '79', content-type: 'applicati
  ► request: XMLHttpRequest {onreadystatechange: null, readyState: 4, timeout: 0, withCredentials: f
    status: 201
    statusText: ""
    ► [[Prototype]]: Object
```

Observe que foi retornado um status 201 de criado com sucesso, poderíamos fazer também um try catch para ficar melhor nosso código.

3 Desafio 1: Faça um formulário de músicas com título, nome do autor, nome da gravadora, ano de lançamento e envie para a API <https://jsonplaceholder.typicode.com/posts>

4 Desafio 2: utilizando uma API pública crie um site para realizar as requisições necessária. Crie no mínimo duas página, uma geral, e uma para detalhar um item em específico.

Sugestões de API's:

OpenWeatherMap API: Fornece dados climáticos e meteorológicos de várias localizações em todo o mundo. Site: <https://openweathermap.org/api>

REST Countries API: Oferece informações sobre países, incluindo detalhes como população, idiomas, bandeiras e mais. Site: <https://restcountries.com/>

CoinGecko API: Fornece dados sobre criptomoedas, incluindo preços, volumes de negociação e outras métricas. Site: <https://coingecko.com/en/api>

PokéAPI: Oferece dados sobre Pokémon, incluindo detalhes sobre espécies, habilidades, tipos e muito mais. Site: <https://pokeapi.co/>

Lembre-se de ler a documentação de cada API para entender como fazer as requisições corretamente.. Fique a vontade para dar a estilização que preferir no projeto, ou também para usar outras API's de seu interesse