

Passo a Passo - REACTJS

Esse passo a passo espera que o frontend (html,css,styled-components) já esteja feito. O objetivo desse documento é referente a comunicação com a API.

Instalar NPM's

```
npm i axios
```

Programar Serviço API

Para comunicar com a API utilizaremos o NPM axios que permite realizar requisições HTTP. Vamos criar uma pasta chamada **/Services** dentro de **/src** e um arquivo chamado **api.js**

A primeira coisa devemos fazer é criar o objeto que apontará para o endereço base da API. Depois, criamos uma classe com as funções referente a API encapsuladas pelas chamadas do axios. A função **api.get(...)** faz uma requisição para um endpoint **GET** localizado no endereço **/chat**, o parâmetro **idSala** da função *listarMensagens* é concatenado ao endereço para ser enviado como parâmetro de rota para a API.

O retorno do *axios* é um objeto complexo, o **JSON** de resposta da API pode ser encontrado no campo **data**.

```
import axios from 'axios'
const api = axios.create({
  baseURL: 'http://localhost:3030'
})

export default class Api {
  async listarMensagens(idSala) {
    let r = await api.get(`/chat/${idSala}`);
    return r.data;
  }
}
```

Programando o Componente

Primeiro, **importamos** a classe criada anteriormente e criamos variável a partir dela.

Depois, criamos a **variável de estado** que irá guardar o JSON retornado pela API.

Por último criamos a função anônima *atualizar* que chama a função *listarMensagens* passando o valor (1) fixo (por enquanto). O retorno da função é enviado como argumento para a função **setChat** que atualizá a variável de estado onde será utilizada para renderização do componente.

```
//...

import { useState } from 'react';
```

```
import Api from '../services/api';
const api = new Api();

export default function Conteudo() {
  const [chat, setChat] = useState([]);

  const atualizar = async () => {
    const mensagens = await api.listarMensagens(1);
    setChat(mensagens)
  }

  return (...)
}
```

A variável de estado **chat** será utilizada na renderização das mensagens conforme código abaixo.

A função **map** converte cada item do array de mensagens retornados da API para um bloco **JSX**. Neste bloco, é utilizada a variável **x** representante de cada mensagem, de forma que sejam utilizados seus campos no **JSX** através das chaves {}, como no caso de **{x.ds_mensagem}**.

```
//..

<div className="chat">
  {chat.map(x =>
    <div>
      <div className="chat-message">
        <div>({new Date(x.dt_mensagem.replace('Z',
        '')).toLocaleTimeString())}</div>
        <div><b>{x.tb_usuario.nm_usuario}</b> fala para <b>Todos</b>:
      </div>
      <div> {x.ds_mensagem} </div>
    </div>
  )}
</div>

//..
```

O arquivo final ficará assim:

```
import { ContainerConteudo } from './conteudo.styled'
import { ChatButton, ChatInput, ChatTextArea } from
'../components/outros/inputs'

import { useState } from 'react';

import Api from '../service/api';
const api = new Api();
```

```

export default function Conteudo() {
  const [chat, setChat] = useState([]);

  const atualizar = async () => {
    const mensagens = await api.listarMensagens(1);
    setChat(mensagens)
  }

  return (
    <ContainerConteudo>
      <div className="container-form"> ... </div>

      <div className="container-chat">

        

        <div className="chat">
          {chat.map(x =>
            <div>
              <div className="chat-message">
                <div>({new Date(x.dt_mensagem.replace('Z',
                '').toLocaleTimeString())}</div>
                <div><b>{x.tb_usuario.nm_usuario}</b> fala para
                <b>Todos</b>:</div>
                <div> {x.ds_mensagem} </div>
              </div>
            </div>
          )}
        </div>
      </div>
    </ContainerConteudo>
  )
}

```