

 devsuperior / sds-dsmovie Public[Code](#)[Issues 1](#)[Pull requests](#)[Actions](#)[Projects](#)[Wiki](#)[Security](#)[Insights](#) main ▾

...

[sds-dsmovie](#) / episodio3 /

acenelio ...

2 days ago



..



README.md

2 days ago

 README.md

Semana Spring React - Episódio 3

Crie um app inédito para seu portfólio com as tecnologias mais demandadas do mercado

Realização

[DevSuperior](#) - Escola de programação



Objetivos do projeto para esta aula

- Integrar back end e front end
- Três pilares do React
 - Componentes
 - Props
 - Estado
- React Hooks
 - useState
 - useEffect
 - useParams

- useNavigate

AVISO: as aulas ficarão disponíveis somente até domingo às 23h59

AVISO: Instruções sobre certificado no Github do [aqui](#)

Passo: Primeira requisição

- Instalar Axios

```
yarn add axios@0.24.0
```

- Definir BASE_URL
- Definir os tipos Movie e MoviePage
- Fazer a requisição

```
export type Movie = {  
  id: number;  
  title: string;  
  score: number;  
  count: number;  
  image: string;  
}  
  
export type MoviePage = {  
  content: Movie[];  
  last: boolean;  
  totalPages: number;  
  totalElements: number;  
  size: number;  
  number: number;  
  first: boolean;  
  numberOfElements: number;  
  empty: boolean;  
}
```

- COMMIT: First request

Passo: React hooks: useState e useEffect

Hooks são funções cujo comportamento está vinculado ao estado e ao ciclo de vida do React a partir de componentes funcionais.

<https://pt-br.reactjs.org/docs/hooks-overview.html>

Hook: `useState`

Manter estado no componente

Hook: `useEffect`

Executar algo na instanciação ou destruição do componente, observar estado

- COMMIT: `useState`, `useEffect`

Passo: Props

Props podem ser entendidas como argumentos do componente React.

<https://pt-br.reactjs.org/docs/components-and-props.html>

NOTA: em uma renderização dinâmica de coleção, cada elemento renderizado DEVE possuir um atributo `key`.

- COMMIT: Props

Passo: `useParams`

- COMMIT: `useParams`

Passo: Mostrar estrelinhas

```
// EX:
// getFills(3.5) => [1, 1, 1, 0.5, 0]
// getFills(4.1) => [1, 1, 1, 1, 0.5]
function getFills(score: number) {

  const fills = [0, 0, 0, 0, 0];

  const integerPart = Math.floor(score);

  for (let i = 0; i < integerPart; i++) {
    fills[i] = 1;
  }

  const diff = score - integerPart;
  if (diff > 0) {
    fills[integerPart] = 0.5;
  }

  return fills;
}
```

- COMMIT: Show score

Passo: Pagination

- Controlar botão habilitado/desabilitado
- Trocar página ao clique do botão

```
const handlePageChange = (newNumber: number) => {  
  setPageNumber(newNumber);  
}
```

- COMMIT: Pagination

Passo: Salvando score, useNavigate

Função para validar email

```
// https://stackoverflow.com/questions/46155/whats-the-best-way-to-validate-an-email-  
export function validateEmail(email: any) {  
  return String(email)  
    .toLowerCase()  
    .match(  
      /^(([^<>()[]\.\,\;\:\s@"]+(\.[^<>()[]\.\,\;\:\s@"]+)*)|("[.+"])(\w[0-9]{1,3}\.|\0  
    );  
}
```

Objeto de configuração da requisição Axios

```
const config: AxiosRequestConfig = {  
  baseUrl: BASE_URL,  
  method: 'PUT',  
  url: '/scores',  
  data: {  
    email: email,  
    movieId: movieId,  
    score: score  
  }  
}
```

- COMMIT: Save, useNavigate

PARABÉNS!



- Quero muito saber seu feedback
 - O que você está achando da nossa abordagem?
 - Você está conseguindo acompanhar?
 - O que você está achando do evento?
 - Participe
 - Comente na página da Semana Spring React
 - Divulgue seu projeto no LinkedIn e marque a DevSuperior
-