

Usando Next.js de maneira simples

sábado, 2 de setembro de 2023


10:19

Instalando React e React Dom com Node

Antes de começar a instalação, crie um arquivo chamado "package.json" e insira e abra chaves ({ }) nele. Depois, no seu terminal, instale o react do seguinte modo:

- Certifique-se de que a instalação vai ocorrer na pasta em que o arquivo "package.json" foi criado: use o comando **cd <path da pasta em que package.json está inserido>**
- Use o comando **npm install react react-dom next** para instalar a biblioteca desejada

Depois disto, verá que o arquivo package.json será atualizado magicamente:



```
{
  "dependencies": {
    "next": "^13.4.19",
    "react": "^18.2.0",
    "react-dom": "^18.2.0"
  }
}
```

Além disto, uma pasta node-modules com mais de 6k arquivos será criada.

Depois disto, podemos copiar o arquivo do exemplo anterior (Getting Started) e apagar algumas coisinhas dele:

1. The `react` and `react-dom` scripts since you've installed them with NPM.
2. The `<html>` and `<body>` tags because Next.js will create these for you.
3. The code that interacts with `app` element and `ReactDOM.render()` method.
4. The `Babel` script because Next.js has a compiler that transforms JSX into valid JavaScript browsers can understand.
5. The `<script type="text/jsx">` tag.
6. The `React.` part of the `React.useState()` function

Como eu já usei um arquivo .js separado para inserir tudo, o que eu precisei fazer foi deletar o HTML criado. Sim, não precisamos dele mais. O que devemos fazer agora é importar o useState para não ter problemas com esse comando na hora de atualizar os likes: `import { useState } from 'react'`. Uma dica: posso mudar a extensão do arquivo para .jsx também.

```

{} package.json U X JS index.js U X
lessons > using-nextjs > JS index.js > 🏠 HomePage
1  import { useState } from 'react'
2
3  function createTitle(title) {
4    return title? title: 'Default title';
5  }
6
7  // Posso passar props, que é um objeto, como um parametro
8  // Ou ainda desestruturá-lo e mostrar somente a title props
9
10 function Header({title}) {
11   // Não posso passar somente "title" entre as tags h1
12   // Devemos utilizar chaves {} para delimitar as vairaveis
13
14   return <h1>{`Trust yourself. ${createTitle(title)}`}</h1>
15 }
16
17 function Paragraph() {
18   return <p><strong>Parágrafo com react:</strong> Lorem ipsum, dolor sit amet consectetur adipisicing elit.
19   Dolorum consequatur sunt non amet quasi rerum nulla, at facere? Nostrum temporibus laboriosam similique,
20   accusantium sequi pariatur! Amet deserunt vero veritatis dolore!</p>
21 }
22
23 function HomePage() {
24   const [likes, setLikes] = useState(0);
25
26   function handleClick() {
27     setLikes(likes + 1);
28   }
29
30   const names = ['Ada Lovelace', 'Grace Hopper', 'Margaret Hamilton'];

```

Agora vamos mudar o index.js para uma pasta chamada "page". Depois disto vamos mudar a declaração da nossa função HomePage para que o Next.js identifique que esta será o principal componente a ser executado.

```

export default function HomePage() {
  const [likes, setLikes] = useState(0);

  function handleClick() {
    setLikes(likes + 1);
  }

  const names = ['Ada Lovelace', 'Grace H

```

O que faremos a seguir é incluir um código (só escrever scripts, que o vscode cria a estrutura) no package.json para criar um ambiente virtual:

```

{} package.json U X JS index.js U
lessons > using-nextjs > {} package.json > ...
1  {
2    > Debug
3    "scripts": {
4      "dev" : "next dev"
5    },
6    "dependencies": {
7      "next": "^13.4.19",
8
9      "react": "^18.2.0",
10
11     "react-dom": "^18.2.0"
12   }
13 }

```

Agora, se você escrever "npm run dev" no seu terminal o arquivo vai ser hospedado em um endereço <http://localhost:3000>. Qualquer mudança, uma vez salva, será refletida na página web.