

FIAP

GRADUAÇÃO

45697056



TDS

Responsive Web Development

Prof. Alexandre Carlos profalexandre.jesus@fiap.com.br

Prof. Luís Carlos lsilva@fiap.com.br





LOCAL-STORAGE

45697056



- Existem duas possibilidades principais para o armazenamento de dados no navegador com ReactJS: `localStorage` e `sessionStorage`.
- O `sessionStorage` é exatamente igual ao `localStorage` exceto pelo fato de os dados salvos são apagados cada vez que o navegador for fechado/recarregado.

- **`localStorage`**

- Qualquer conteúdo / dado salvo no objeto `localStorage` estará disponível depois que o navegador for reiniciado (fechado e aberto novamente).





Métodos do objeto localStorage

45697056
■ ■ ■

Os métodos fornecidos pelo objeto localStorage são os seguintes:

- setItem
- getItem
- removeItem
- clear





Utilizando o localStorage

45697056
■ ■ ■

Para salvar alguma informação no localStorage, você pode usar o método `setItem()`. Este método deve receber uma chave e um valor.

```
localStorage.setItem("mykey", "myvalue");
```





Utilizando o localStorage

45697056



Para recuperar alguma informação no localStorage, você precisa usar o método getItem. O método getItem deve receber a chave dos dados que você deseja recuperar:

```
localStorage.removeItem("mykey");
```





Utilizando o localStorage

45697056



Você pode remover um item do localStorage usando o método `removeItem()`. Neste método deve ser entregue a chave do item a ser removido:

```
localStorage.removeItem("mykey");
```





Utilizando o localStorage

45697056
■ ■ ■

Para limpar tudo do localStorage, você deve usar o método `clear()` no objeto `localStorage`:

```
localStorage.clear();
```





Salvando matrizes no localStorage

45697056

■ ■ ■

Além de valores únicos, você também pode salvar uma matriz no localStorage.

Neste exemplo, temos uma matriz com números:

```
var ourArray =[1,2,3,4,5];
```

Agora podemos salvá-lo em localStorage usando o setItem() método:

```
localStorage.setItem("ourarraykey",JSON.stringify(ourArray));
```



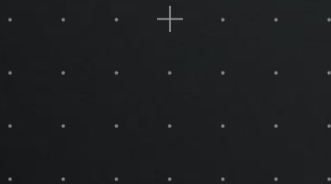
Salvando matrizes no localStorage

45697056
■ ■ ■

Para ser salvo, a matriz deve primeiro ser convertida em uma string. No exemplo mostrado acima, estamos usando o método `JSON.stringify` para fazer isso.

Ao recuperar nossos dados de `localStorage`, devemos convertê-los de volta em uma matriz:

```
var storedArray = localStorage.getItem("ourarraykey");  
ourArray = JSON.parse(storedArray);
```





Exemplo de Componente React

Veja abaixo um exemplo bem prático com React.

Neste exemplo, toda vez que o usuário inserir algum valor no input seu valor será salvo no localStorage.

```
import React from 'react';

const App = () => {
  const [value, setValue] = React.useState('');

  const onChange = event => {
    localStorage.setItem('myValueInLocalStorage', event.target.value);

    setValue(event.target.value);
  };

  return (
    <div>
      <h1>Hello React with Local Storage!</h1>

      <input value={value} type="text" onChange={onChange} />

      <p>{value}</p>
    </div>
  );
};

export default App;
```



React

Exemplo Simplificado arquivo JS

Veja abaixo um exemplo bem prático com Javascript.

Neste exemplo, toda vez que o form for submetido os dados dos inputs serão salvos no localStorage com.

```
<div>
  <form action="index.html" method="get" onsubmit="return true">
    <h2>ENTRAR</h2>
    <div>
      <label for="idUser">Usuario</label>
      <input type="text" name="txtUser" id="idUser" placeholder="Usuário">
    </div>
    <div>
      <label for="idEmail">Email</label>
      <input type="email" name="txtEmail" id="idEmail" placeholder="Email">
    </div>
    <div>
      <button type="submit" id="btnSubmit">Cadastrar</button>
    </div>
  </form>
</div>
```

```
92
93 //Lista para armazenar todos os usuários:
94 let listaDeUsuarios = [];
95 //Objeto Usuário base
96 function Usuario(nome,email) {
97   this.nome = nome;
98   this.email = email;
99 }
100
101 const botaoEnviar = document.querySelector("#btnSubmit");
102
103 botaoEnviar.addEventListener("click", ()=>{
104   const inputUser = document.querySelector("#idUser");
105   const inputEmail = document.querySelector("#idEmail");
106
107   let usuario = new Usuario(inputUser.value,inputEmail.value);
108   listaDeUsuarios.push(usuario);
109   localStorage.setItem("users",listaDeUsuarios);
110 });
111
112
113
114
115
116
```

Lista de usuarios

Objeto base

Botao disparador

inputs

Evento do botao

instancia do objeto recebendo os dados dos campos

Lista de usuarios recebendo o obj usuario.

LocalStorage armazenando a lista de usuarios.



Exemplo Simplificado arquivo JS

45697056



```
<div>
  <form action="index.html" method="get" onSubmit="return true">
    <h2>ENTRAR</h2>
    <div>
      <label for="idUser">Usuario</label>
      <input type="text" name="txtUser" id="idUser" placeholder="Usuário">
    </div>
    <div>
      <label for="idEmail">Email</label>
      <input type="email" name="txtEmail" id="idEmail" placeholder="Email">
    </div>
    <div>
      <button type="submit" id="btnSubmit">Cadastrar</button>
    </div>
  </form>
</div>
```



React

Exemplo Simplificado arquivo JS

45697056

■ ■ ■

```
92
93 //Lista para armazenar todos os usuários:
94 let listaDeUsuarios = [];
95 //Objeto Usuário base
96 function Usuario(nome,email) {
97     this.nome = nome;
98     this.email = email;
99 }
100
101 const botaoEnviar = document.querySelector("#btnSubmit");
102
103 botaoEnviar.addEventListener("click", ()=>{
104     const inputUser = document.querySelector("#idUser");
105     const inputEmail = document.querySelector("#idEmail");
106
107     let usuario = new Usuario(inputUser.value,inputEmail.value);
108     listaDeUsuarios.push(usuario);
109     localStorage.setItem("users",listaDeUsuarios);
110 });
```

Lista de usuarios

Objeto base

Botao disparador

inputs

Evento do botao

instancia do objeto
recebendo os dados dos campos

Lista de usuarios recebendo o obj usuario.

LocalStorage armazenando a
lista de usuarios.



Concluindo

45697056
■ ■ ■

Viu como é fácil salvar dados no navegador com React e localStorage?

Salvar informações no localStorage é muito útil para guardar informações de login de usuários em nossas aplicação, como por exemplo, gravar o token do usuário que será usado ao longo da utilização da aplicação.



DUVIDAS





Copyright © 2015 - 2023 Prof. Luís Carlos S. Silva
Prof. Alexandre Carlos de Jesus

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).