

Programação & Serviços Web

React – Aula 4

2019/2020

FORMULÁRIOS

Formulários

- Usado em websites para
 - Login
 - Submeter informação (fazer encomendas, comprar um bilhete, etc)

Formulários

- Em Html estamos habituados a:
 - `<form>...</form>`
 - E, dentro do form
 - `<input>`
 - `<textarea>`
 - `<button>`
 - `<select>`
 - Entre outros
- Em react, funciona exatamente da mesma forma

Única Diferença

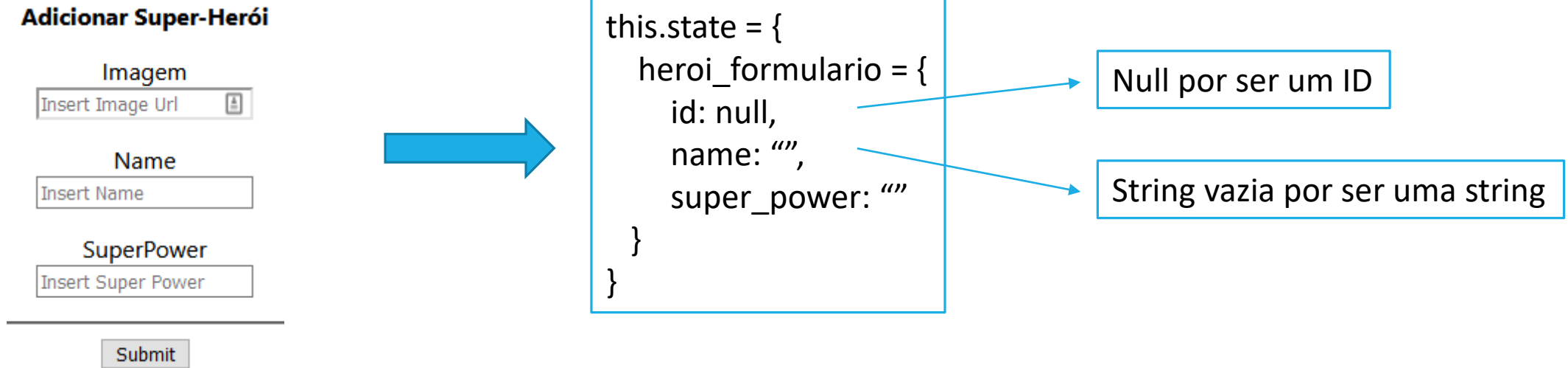
- Num formulário em HTML fazemos a submissão via AJAX (ou equivalente).
- Em React (e falando neste caso em questão), iremos atualizar a nossa lista de heróis (o state, e não o ficheiro)
- O React é uma aplicação de Frontend, pelo que não podemos actualizar ficheiros que sejam usados para carregar dados (o heroes.js nunca poderá ser actualizado via React)

Processo de criação de Forms

- Existem diversas formas de criar Formulários no React. Esta é apenas uma sugestão de como o fazer de forma simples:
- 1) Criar um componente de Classe
- 2) Criar no render o código necessário para o formulário

Processo de criação de Forms

3) No state deste novo componente, criar um objeto com os campos presentes no form. Ex:



Processo de criação de Forms

- 4) Nos inputs do formulário, adicionar o evento onChange, de forma a atualizar o state do componente

```
<p>
  <label
htmlFor="name">Nome</label><br />
    <input type="text" id="name"
name="name" placeholder="Insert Name"
onChange={this.updateField, "name"} />
</p>
```

```
updateField = (event, fieldName) => {
  var newHeroi = this.state.heroi_formulario;
  newHeroi[fieldName] = event.target.value;
  this.setState({
    heroi_formulario : newHeroi
  })
  event.preventDefault();
}
```

Valor do input



Processo de criação de Forms

- 5) Adicionar o evento **onSubmit** do formulário, chamando uma função e enviar o state

```
<form onSubmit={(e) => this.props.submit(e, this.state.heroi_formulario)}>
```

Visto que a nossa lista de heróis está presente no componente **Content**, a função que irá gravar o novo elemento **deve estar lá também**.

Envia-se para o componente **Formulário** a função, que é acedida neste através do props

Processo de criação de Forms

```
submitForm = (e, novo_heroi) => {  
  e.preventDefault();  
  var list = this.state.list_of_heroes;  
  list.push({  
    id: list[list.length - 1].id + 1,  
    name: novo_heroi.name,  
    super_power: novo_heroi.super_power,  
    image: novo_heroi.image  
  });  
  this.setState({  
    list_of_heroes: list  
  });  
  this.props.history.push('/dashboard');  
}
```

Faz um redirect para a página /dashboard

Para que este **redirect funcione**, é necessário:

- Fazer o *import* da classe **withRouter** do **react-router-dom**
- Atualizar o *export* do componente em questão (**Content**)

Para o seguinte:

`export default withRouter(Content);`

Em vez de:
`export default Content;`



Processo de criação de Forms

- Para finalizar, basta adicionar a Route do /dashboard/add de forma a chamar o Componente do Formulário

```
<Route path="/dashboard/add" render={props =>  
  (<Formulárioi {...props} submit={this.submitForm} />)  
} />
```

Edição de super-heróis

- No final de ter o formulário de adição de super-heróis, usa-se o mesmo formulário para fazer a edição dos mesmos.
- 1) Criar a route para a edição, não esquecendo do token. Por exemplo: /dashboard/edit/:id
- 2) Enviar a lista de heróis para o Formulário (para fazer a associação entre o ID)

```
<Route path="/dashboard/edit/:id" render={props =>
  (<SuperForm {...props} list_of_heroes={this.state.list_of_heroes} submit={this.submitForm} />)
} />
```

Edição de super-heróis

- 3) No construtor do componente Formulário, verifique se o seu url tem algum ID (através do props.match). Se existir, é porque se trata de uma edição. Caso contrário, é porque é uma adição.
- 4) No caso de haver o token ID, procure na lista de heróis pelo herói com o ID em questão. Se existir, atualize a definição do state com estes valores.

```
this.state = {  
  element: element !== undefined ? element : {  
    id: null,  
    name: "",  
    super_power: "",  
    image: ""  
  },  
}
```

Edição de super-heróis

- 5) Atualize o seu formulário para que o evento “value” dos inputs tenha o valor definido no state:

```
<input type="text" id="name" name="name" placeholder="Insert Name" onChange={this.updateField}  
value={this.state.super_heroi.name} />
```

- 6) Atualize a **função de submissão** presente no componente **Content** para que no caso de o elemento enviado ter um ID, actualize a lista
- Caso contrário, adiciona o super herói.

Formulários não controlados

- Pode ser entediante usar formulários controlados (para cada input ter que adicionar o event do onChange, em que o que varia é o tipo)
- Isto pode ser ainda pior quando se está a converter um projeto para React, ou integrar o React numa outra aplicação. Para estas situações, existem os **Formulários não Controlados**.
- Nesta UC não iremos abordar este conteúdo. No entanto, está disponível na documentação do React.

Ficha 4

Leitura Complementar

- <https://reactjs.org/docs/forms.html>