

Mãos na Massa

Chegou a hora de você pôr em prática o que foi visto na aula. Para isso, execute os passos listados abaixo.

1 - Crie um arquivo `PopUp.js` e `FormValidator.js` e importe no seu componente Formulário

```
import FormValidator from './FormValidator';
import PopUp from './PopUp';
```

2 - Corrija a criação do **FormValidator** no construtor do componente Formulário.

//DENTRO DO CONSTRUTOR.

```
this.validador = new FormValidator([
  {
    campo: 'nome',
    metodo: 'isEmpty',
    validoQuando: false,
    mensagem: 'Entre com um nome'
  },
  {
    campo: 'livro',
    metodo: 'isEmpty',
    validoQuando: false,
    mensagem: 'Entre com um livro'
  },
  {
    campo: 'preco',
    metodo: 'isInt',
    args: [{min: 0, max: 99999}],
    validoQuando: true,
    mensagem: 'Entre com um valor numérico'
  }
]);

this.stateInicial = {
  nome: '',
  livro: '',
  preco: '',
  validacao: this.validador.valido()
}
```

3 - Modifique o método **submitFormulario** conforme abaixo:

```
submitFormulario = () => {
  const validacao = this.validador.valida(this.state);
  console.log(validacao)
  if(validacao.isValid){
    this.props.escutadorDeSubmit(this.state);
    this.setState(this.stateInicial);
  }
}
```

```

    }else {
      const { nome, livro, preco } = validacao;
      const campos = [nome, livro, preco];
      const camposInvalidos = campos.filter(elem => {
        return elem.isInvalid
      });
      camposInvalidos.forEach(campo => {
        PopUp.exibeMensagem('error', campo.message);
      });
    }
  }
}

```

4- Abra seu arquivo **FormValidator** crie sua classe e adicione os métodos **valida** e **valido**:

```

import validador from 'validator';

class FormValidator {

  constructor(validacoes) {

    // validacoes é um array específico do formulário com regras de validação
    this.validacoes = validacoes;
  }

  valida(state) {

    //itera pelo array de regras de validação e constrói
    //um objeto validacao e retorna-o

    //começa assumindo que está tudo válido
    let validacao = this.valido();

    this.validacoes.forEach(regra => {

      //Se o campo não tiver sido marcado anteriormente como invalido por uma regra.
      if (!validacao[regra.campo].isInvalid) {
        //Determina o valor do campo, o método a ser invocado
        //e os argumentos opcionais pela definição da regra
        const campoValor = state[regra.campo.toString()];
        const args = regra.args || [];
        const metodoValidacao = typeof regra.metodo === 'string' ?
          validador[regra.metodo] : regra.metodo;

        if (metodoValidacao(campoValor, ...args, state) !== regra.validoQuando) {
          validacao[regra.campo] = {
            isInvalid: true,
            message: regra.mensagem
          };
          validacao.isValid = false;
        }
      }
    });
  }
}

```

```
    });  
    return validacao;  
  }  
  //cria um objeto validacao para um form válido  
  valido() {  
    const validacao = {};  
  
    this.validacoes.map(regra => (  
      validacao[regra.campo] = { isInvalid: false, message: '' }  
    ));  
  
    return { isValid: true, ...validacao };  
  }  
  
}  
  
export default FormValidator;
```

5 - Abra seu arquivo PopUp.js e adicione o código abaixo:

```
import M from 'materialize-css';  
  
const PopUp = {  
  exibeMensagem: (status, msg) => {  
    if (status === "success")  
      M.toast({ html: msg, classes: "green", displayLength: 2000 })  
  
    if (status === "error")  
      M.toast({ html: msg, classes: "red", displayLength: 2000 })  
  
  }  
}  
  
export default PopUp;
```

6 - Teste no navegador e veja se tudo funciona!