



---

# React Form

Khóa học: ReactJS

# Mục tiêu

---



- Xử lý được dữ liệu trong Form
- Sử dụng được Submitting Forms
- Sử dụng được Multiple Input Fields
- Sử dụng được Validating Form Input
- Xử lý được Error Message
- Sử dụng được thẻ nhập liệu căn bản Textarea, Select, Input
- Upload được file
- Sử dụng Formik

- Giống như HTML, React sử dụng form để cho phép người dùng tương tác với trang web.
- Form được sử dụng để tìm kiếm , nhập thông tin liên lạc , login ...
- Các phần tử trong form tương đối đa dạng.
- Một số phần tử thông dụng như input, checkbox, radio, submit button...



# Thêm Form trong React

---

Thêm một form với React có cấu trúc tương tự như viết HTML.

## Ví dụ:

```
import React from 'react';
const MyForm = () => {
  return (
    <form>
      <h1>Hello</h1>
      <p>Enter your name:</p>
      <input type="text"/>
    </form>;)
}
export default MyForm;
```



# Xử lý dữ liệu của Form

---

- Là cách xử lý dữ liệu khi giá trị được thay đổi hoặc đã gửi đi
- Trong HTML, dữ liệu của form được quản lý bởi DOM
- Trong React, dữ liệu của form được quản lý bởi các component
- Khi dữ liệu được xử lý bởi các component, tất cả dữ liệu được lưu trữ ở state
- Chúng ta có thể kiểm soát các thay đổi bằng cách thêm trình xử lý sự kiện trong thuộc tính onChange



---

# Demo

Xử lý dữ liệu của Form

# Cập nhật trạng thái



Khi cập nhật trạng thái mới, chúng ta phải sử dụng `this` để có thể bind tới phương thức **`handleChange`**

## Ví dụ:

```
import React, {useState} from 'react';
function MyForm () =>{
  const [username, setUsername] = useState('');
  const handleChange = event => setUsername(event.target.value)
  return (
    <div>
      <form>
        <h1>Hello {username}</h1>
        <p>Enter your name:</p>
        <input type='text' value={username} onChange={
handleChange }
        />
      </form>
    </div>
  )}
```

# Điều kiện render (Conditional Rendering)



Xét ví dụ dưới đây và lưu ý những điều sau:

- Tạo ra một biến trống, gọi là header
- Thêm câu lệnh if để chèn nội dung vào biến header, nếu người dùng đã thực hiện bất kỳ thao tác nhập nào.
- Chèn biến tiêu đề vào đầu ra, sử dụng dấu ngoặc nhọn.





---

# Demo

Điều kiện render (Conditional Rendering)

# Submitting Forms

---



Chúng ta có thể kiểm soát hành động gửi bằng cách thêm trình xử lý sự kiện trong thuộc tính **onSubmit**



---

# Demo

Submitting Forms



# Multiple Input Fields

---

- Bạn có thể kiểm soát các giá trị của nhiều đầu vào bằng cách thêm thuộc tính tên vào mỗi phần tử.
- Khi bạn khởi tạo trạng thái trong hàm tạo, hãy sử dụng tên
- Để truy cập các trình xử lý sự kiện, hãy sử dụng cú pháp `event.target.name` và `event.target.value`.
- Để cập nhật trạng thái trong phương thức `this.setState`, hãy sử dụng dấu ngoặc vuông [ký hiệu dấu ngoặc vuông] xung quanh tên thuộc tính.



---

# Demo

Multiple Input Fields

# Tổng kết (1)

---



Qua nội dung trên, chúng ta đã tìm hiểu :

- Viết Form trong React
- Xử lý dữ liệu trong form
- Điều kiện render của Form
- Viết Submit Form và Multiple Input Fields

# Thẻ nhập liệu căn bản Textarea, Select...

---



Trong nội dung này, chúng ta sẽ tìm hiểu các phần tử sau:

- TextArea
- Select
- Input File

# Thẻ nhập liệu Textarea

---



Phần tử textarea trong React khác so với thẻ HTML thông thường. Trong HTML, giá trị của một textarea là văn bản giữa thẻ bắt đầu `<textarea>` và thẻ kết thúc `</textarea>`

Ví dụ :

```
<textarea>
```

```
  Hello there, this is some text in a text area
```

```
</textarea>
```





# Thẻ nhập liệu Textarea

---

Trong React, giá trị của một textarea được đặt trong một thuộc tính giá trị:

**Ví dụ :**

```
import React, {useState} from 'react'
function Form() {
  const [state, setState] = useState({
    description: 'The content of a textarea goes
in the value attribute'
  })
  return (
    <form><textarea value={state.description}
/></form>
  )
}
```

# Thẻ nhập liệu Select trong HTML

---



Trong HTML, giá trị đã chọn trong danh sách drop down được xác định bằng thuộc tính **selected**:

Ví dụ:

```
<select>  
  <option value="Ford">Ford</option>  
  <option value="Volvo" selected>Volvo</option>  
  <option value="Fiat">Fiat</option>  
</select>
```



# Thẻ nhập liệu Select trong React

---

Trong React, giá trị đã chọn được xác định bằng thuộc tính value trên thẻ select.

## Ví dụ:

```
import React, {useState} from 'react'
function Form() {const [state, setState] = useState({mycar: 'Volvo'})
  return (
    <form>
      <select value={state.mycar}>
        <option value="Ford">Ford</option>
        <option value="Volvo">Volvo</option>
        <option value="Fiat">Fiat</option>
      </select>
    </form>
  );
}
```



# Thẻ nhập liệu Input File

---

Trong HTML, `<input type = "file">` cho phép người dùng chọn một hoặc nhiều tệp từ bộ nhớ thiết bị để tải lên máy chủ hoặc được JavaScript thao tác thông qua API:

```
<input type="file" />
```

Vì giá trị của nó là chỉ đọc nên nó là một thành phần không được kiểm soát trong React.

Trong React, `<input type = "file" />` luôn là một thành phần không được kiểm soát vì giá trị của nó chỉ có thể được đặt bởi người dùng chứ không phải theo lập trình. Chúng ta nên sử dụng File API để tương tác với các tệp tin.



# Thẻ nhập liệu Input File

---

Ví dụ cấu trúc file ảnh:

```
{
  lastModified: 1588350162449
  lastModifiedDate: Fri May 01 2020 17:22:42
  GMT+0100 (British Summer Time) {} // Date object
  name: "Pluralsight_logo.png"
  size: 68317
  type: "image/png"
  webkitRelativePath: ""
  __proto__: File
}
```



---

# Demo

Thẻ nhập liệu Input File

# Tổng kết (2)

---



Qua nội dung vừa rồi, chúng ta đã tìm hiểu:

- Sử dụng thẻ nhập liệu TextArea
- Sử dụng thẻ Select
- Sử dụng thẻ Input Upload để upload file

# Form validation trong React

---



- Trong React, Form validation cho phép hiển thị thông báo lỗi nếu người dùng điền không chính xác vào biểu mẫu với loại đầu vào mong đợi.
- Có nhiều cách xác thực các biểu mẫu trong React, tuy nhiên cách ngắn gọn là tạo ra các hàm validation với các quy tắc xác thực
- Bạn có thể xác thực đầu vào biểu mẫu khi người dùng đang nhập hoặc bạn có thể đợi cho đến khi biểu mẫu được gửi.



---

# Demo (1)

Nội dung: Khi điền thông tin tuổi, người dùng sẽ nhận được cảnh báo nếu field tuổi không phải là số.

# Demo (2)

Nội dung: Validation được thực hiện khi biểu mẫu được gửi thay vì khi người dùng nhập vào.



# Error Message trong validation

---

Form validation trong React cho phép hiển thị thông báo lỗi nếu người dùng điền không chính xác vào biểu mẫu khớp với loại đầu vào mong đợi.

**Ví dụ :** Khi bạn điền tuổi của mình không phải là số, một thông báo lỗi sẽ hiển thị



---

# Demo

# Tổng kết (3)

---



Qua nội dung vừa rồi, chúng ta đã tìm hiểu:

- Tìm hiểu và cách xây dựng Form Validation
- Thêm Error Message khi flined nhập vào không hợp lệ



# Giới thiệu Formik

---

- Khi làm việc với các thư viện về front-end như (ReactJS, VueJS hay Angular) thì việc quản lý các form là điều không thể thiếu
- Quản lý form là cách mà quản lý thông tin:
  - Người dùng nhập vào, tiến hành kiểm tra dữ liệu đầu vào đó (validate)
  - Hiển thị kết quả cho người dùng nếu có hoặc gửi (submit) lên server nếu không có thông tin người dùng là hợp lệ
- Trong React, sử dụng thư viện Formik sẽ giúp việc quản lý form trở nên đơn giản hơn

# Formik là gì ?

---



Formik là một thư viện giúp chúng ta giải quyết các vấn đề chính sau:

- Nhận giá trị trong và ngoài state form
- Validate giá trị và đưa ra lỗi
- Xử lý việc submit form

# Sử dụng Formik

---



Cài đặt

```
npm install formik --save
```

hoặc

```
yarn add formik
```

Sau đó import vào component:

```
import { Formik } from 'formik';
```





# Validation trong Formik

---

Có 2 cách xác thực biểu mẫu trong Formik:

1. `<Formik validate>` và `withFormik ({validate: ...})`
2. `<Formik validationSchema>` và `withFormik ({validationSchema: ...})`



# withFormik() trong Formik

---

## validate

<Formik> và withFormik() nhận một prop / option được gọi là validate chấp nhận một hàm đồng bộ (synchronous function) hoặc không đồng bộ (asynchronous function).



---

# Demo

withFormik() trong Formik



# validationSchema trong Formik

---

Chúng ta sử dụng thư viện bên thứ 3 là Yup để xác thực. Formik có tùy chọn cấu hình / hỗ trợ đặc biệt cho Yup để chuyển đổi các lỗi xác thực của Yup thành một đối tượng đẹp có key khớp với các giá trị.

Cài đặt Yup vào dự án:

```
npm install yup -save
```



---

# Demo

validationSchema trong Formik



# Thời điểm validate

---

Chúng ta có thể kiểm soát thời điểm Formik validate bằng cách thay đổi các giá trị sau:

- `<Formik validateOnChange>`
- `<Formik validateOnBlur>`

Tùy thuộc vào nhu cầu của sản phẩm, chúng ta sẽ chọn thời điểm phù hợp.



# Validation trong Formik (1)

---

Theo mặc định, Formik sẽ chạy các phương thức validation như sau:

- **Sau khi thay đổi sự kiện, phương thức (những thứ có giá trị cập nhật)**
  - handleChange
  - setFieldValue
  - setValues
- **Sau khi các sự kiện / phương thức bị làm mờ (những thứ đã cập nhật được chạm vào)**
  - handleBlur
  - setTouched
  - setFieldTouched



# Validation trong Formik (2)

---

- **Bất cứ khi nào gửi submit**  
    handleSubmit  
    submitForm
- **Ngoài ra còn có các phương thức trợ giúp được cung cấp để xác thực**  
    validateForm  
    validateField



# Hiển thị Error Messages

---



Error messages phụ thuộc vào validation .Nếu có lỗi xảy ra, hoặc hàm validation tạo ra đối tượng lỗi khớp các giá trị / giá trị ban đầu thì các trường lỗi có thể truy cập từ đối tượng lỗi



---

# Demo

Hiển thị Error Messages

# Giao diện Formik

---



Tạo một form đơn giản với Formik:

Ta import React, react-dom, và các component của Formik

```
import React from 'react';
```

```
import ReactDOM from 'react-dom';
```

```
import { Formik, Field, Form } from 'formik';
```

Sau đó ta sẽ bắt đầu viết một cấu trúc Form dựa trên thư viện Formik



# Demo

Giao diện Formik

## Đăng kí thông tin

Name

Phone

Address

# Tổng kết

---



Qua bài học này, chúng ta đã tìm hiểu:

- Xử lý được dữ liệu trong Form
- Sử dụng được Submitting Forms
- Sử dụng được Multiple Input Fields
- Sử dụng được Validating Form Input
- Xử lý được Error Message
- Sử dụng được thẻ nhập liệu căn bản Textarea, Select, Input
- Upload được file
- Sử dụng Formik

# Thực hành

# [Thực hành] Form đăng ký tài khoản mới



## Mục tiêu

Tạo được dự án React JS

Thao tác với state thông qua event

Thực hành validate form

## Mô tả bài toán

Viết chương trình cho phép sign up

## Sign up

Username

Email

Password

Confirm password

Submit

# [Thực hành] Validate Sign up form



## Mục tiêu

- Tạo được dự án React JS
- Thao tác với state thông qua event
- Thực hành với cơ chế condition rendering
- Thực hành validate form

## Mô tả bài toán

Viết chương trình cho phép sign up

### Sign up

Username

Username error

Email

Email error

Password

Password error

Confirm password

Confirm password error



# [Thực hành] Validate Login form

---



## Mục tiêu

- Tạo được dự án React JS
- Thao tác với state thông qua event
- Thực hành validate form

## Mô tả bài toán

Viết chương trình cho phép Login

## Login

Email

Password

Submit

# [Thực hành] Validate Login form bằng Formik



## Mục tiêu

- Tạo được dự án React JS
- Thao tác với state thông qua event
- Thực hành validate form bằng Formik

## Mô tả bài toán

Viết chương trình cho phép Login

## Login

Email

Password

Submit

# [Bài tập] Validate Contact form bằng Formik



## Mục tiêu

- Tạo được dự án React JS
- Thao tác với state thông qua event
- Thực hành validate form bằng Formik

## Mô tả bài toán

Viết chương trình cho phép tạo Contact form

### Contact form

Name

Email

Phone

Message

Submit

# [Bài tập] Ứng dụng quản lý sách



## Mục tiêu

- Tạo được dự án React JS
- Thao tác với state thông qua event
- Thực hành validate form bằng Formik

## Mô tả bài toán

- Viết chương trình quản lý thư viện

### Library

Tiêu đề

Số lượng

Submit

Title	Number	Actions
book1	1	<button>Edit</button> <button>Delete</button>
book2	2	<button>Edit</button> <button>Delete</button>
book3	2	<button>Edit</button> <button>Delete</button>

# [Bài tập] Validate Mail form bằng Formik



## Mục tiêu

- Tạo được dự án React JS
- Thao tác với state thông qua event
- Thực hành validate form bằng Formik

## Mô tả bài toán

- Viết chương trình quản lý soạn thảo và gửi mail

## Mail form

To

Title

Message

Choose File

No file chosen

Submit

# [Bài tập] Validate Covid-19 form bằng Formik



## Mục tiêu

- Tạo được dự án React JS
- Thao tác với state thông qua event
- Thực hành validate form bằng Formik

## Mô tả bài toán

- Viết chương trình khai báo y tế

## Tờ khai y tế

Họ tên

Số hộ chiếu /CMND

Năm sinh

Giới tính ☒ Nam ☐ Nữ

Quốc tịch

Công ty làm việc

Bộ phận làm việc

Có thẻ bảo hiểm y tế ☐

**Địa chỉ liên lạc tại Việt Nam**

Tỉnh thành