

Form handling

Tìm hiểu thư viện formik



CYBERSOFT
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH



- ❖ Ở các slide đầu, mình đã hướng dẫn các bạn xử lý form thông qua state và handle sự kiện **onChange**. Ở phần này, chúng ta sẽ xử lý form thông qua 1 thư viện khá phổ biến hiện tại, có tên là Formik, hỗ trợ xử lý form và validation form.
- ❖ Cách sử dụng:
 - Cài đặt formik vào dự án: ***npm install formik -S***
 - Bài tập demo : xử lý form đăng nhập với formik

Class component

Xét giá trị mặc định cho form

- ❖ Trong một số chức năng, ví dụ như update User, các bạn sẽ cần phải fill thông tin của user đó lên form để cập nhật, v làm sao làm được điều đó với formik?



```
class SignInScreen extends Component {  
  render() {  
    return (  
      <div className="container">  
        <div className="row">  
          <div className="col-5 mx-auto">  
            <Formik  
              initialValues={{  
                taiKhoan: "trungthieu",  
                matKhai: "123"  
              }}  
              onSubmit={values => {  
                console.log(values);  
              }}  
              render={({ handleChange, values }) => (  
                <Form>  
                  <h4 className="display-4">Đăng Nhập</h4>  
                  <div className="form-group">  
                    <label htmlFor="taiKhoan">Tên Tài Khoản</label>  
                    <input  
                      type="text"  
                      name="taiKhoan"  
                      onChange={handleChange}  
                      value={values.taiKhoan}  
                      className="form-control"  
                    />  
                  </div>  
                  <div className="form-group">  
                    <label htmlFor="matKhai">Mật Khẩu</label>  
                    <input  
                      type="password"  
                      name="matKhai"  
                      onChange={handleChange}  
                      value={values.matKhai}  
                    />  
                  </div>  
                </Form>  
              )  
            </div>  
          </div>  
        </div>  
      </div>  
    );  
  }  
}
```

ở đây mọi người có thể xét giá trị mặc định cho initialValues như bên, hoặc có thể gán initialValues={object của các bạn},

Ngoài handleChange, ta có thể lấy thêm được values, chính là giá trị mà formik đang

đang

Validation with formik and yup



CYBERSOFT
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH



- ❖ Yup là một thư viện hỗ trợ chúng ta validation form , các bạn có thể sử dụng nó với JS thuần, angular, react...
- ❖ Yup và formik là bộ đôi kết hợp khá ăn ý, giúp việc validation và hiển thị error message sẽ dễ dàng hơn rất nhiều
- ❖ Bản chất của **yup** là giúp ta tạo ra 1 **schema** (bản mẫu) , ví dụ như bản mẫu có bao nhiêu thuộc tính, mỗi thuộc tính có required hay không, có minlength hay maxlength gì không... Sau đó ta sẽ lấy object ta muốn validation so sánh với **schema** , nếu khớp thì là hợp lệ, còn không thì ngược lại.
- ❖ Cài đặt yup : ***npm i yup -S***
- ❖ Document: <https://github.com/jquense/yup>

Validation with formik and yup

- ❖ Tạo ra 1 schema để validation form đăng nhập

```
import * as yup from "yup";

const schema = yup.object().shape({
  taiKhoan: yup
    .string()
    .required("Vui lòng nhập tài khoản")
    .matches(/^[A-Za-z]+$/, "Tài khoản không đúng định dạng"),
  matKhau: yup
    .string()
    .required("Vui lòng nhập mật khẩu")
    .min(6, "Mật khẩu phải tối thiểu 6 kí tự")
    .max(18, "Mật khẩu phải tối đa 18 kí tự")
});
```

- ❖ Áp dụng vào formik: khi ta nhấn submit, formik sẽ tự check, nếu đúng schema mới được submit.

```
<div className="row">
  <div className="col-5 mx-auto">
    <Formik
      validationSchema={schema}
      initialValues={{
        taiKhoan: "trunghieuv",
        matKhau: "123"
      }}
      onSubmit={values => {
        console.log(values);
      }}
      render={({ handleChange, values }) => (
```

Validation with formik and yup



- ❖ Hiển thị error message ra màn hình
- ❖ Lưu ý : chuyển input thành component Field của Formik để sử dụng được đối tượng touched

```
render={({ handleChange, values, errors, touched }) => (  
  <Form>  
    <h4 className="display-4">Đăng Nhập</h4>  
    <div className="form-group">  
      <label htmlFor="Tài Khoản">Tài Khoản</label>  
      <Field  
        type="text"  
        name="taiKhoan"  
        onChange={handleChange}  
        value={values.taiKhoan}  
        className="form-control"  
      />  
      {errors.taiKhoan && touched.taiKhoan ? <div>{errors.taiKhoan}</div> : ''}  
    </div>  
    <div className="form-group">  
      <label htmlFor="Mật Khẩu">Mật Khẩu</label>  
      <Field  
        type="password"  
        className="form-control"  
        name="matKhau"  
        onChange={handleChange}  
        value={values.matKhau}  
      />  
      {errors.matKhau && touched.matKhau ? <div>{errors.matKhau}</div> : ''}  
    </div>  
    <div className="form-group text-center">  
      <button type="submit" className="btn btn-success">  
        Đăng nhập  
      </button>  
    </div>  
  </Form>  
)}
```

Function component

Validation with formik and yup

- ❖ Tạo ra 1 schema để validation form đăng nhập

```
import * as yup from "yup";

const schema = yup.object().shape({
  taiKhoan: yup
    .string()
    .required("Vui lòng nhập tài khoản")
    .matches(/^[A-Za-z]+$/, "Tài khoản không đúng định dạng"),
  matKhau: yup
    .string()
    .required("Vui lòng nhập mật khẩu")
    .min(6, "Mật khẩu phải tối thiểu 6 kí tự")
    .max(18, "Mật khẩu phải tối đa 18 kí tự")
});
```

- ❖ Áp dụng vào formik: khi ta nhấn submit, formik sẽ tự check, nếu đúng schema mới được submit.

Validation with formik and yup

```
const SignInScreen = () => {  
  const { handleChange, values } = useFormik({  
    initialValues: {  
      taiKhoan: "",  
      matKhai: "",  
    },  
    validationSchema: schema,  
  });  
  return (  
    <div>  
      <form>  
        <h1>Đăng nhập</h1>  
        <div className="form-group">  
          <label htmlFor="">Tài khoản</label>  
          <input  
            type="text"  
            name="taiKhoan"  
            onChange={handleChange}  
            value={values.taiKhoan}  
            placeholder="tài khoản"  
          />  
        </div>  
        <div className="form-group">  
          <label htmlFor="">Mật khẩu</label>  
          <input  
            type="text"  
            name="matKhai"  
            value={values.matKhai}  
            onChange={handleChange}  
            placeholder="mật khẩu"  
          />  
        </div>  
      </form>  
    </div>  
  );  
};
```

set values for form

```
1 import React from "react";
2 import { useFormik } from "formik";
3
4 const SignInScreen = () => {
5   const { handleChange, values, setValues } = useFormik({
6     initialValues: {
7       taiKhoan: "",
8       matKhai: "",
9     },
10   });
11
12   const setDefaultValue = () => {
13     const defaultAccount = {
14       taiKhoan: "hieudeptrai",
15       matKhai: "12712839",
16     };
17     setValues(defaultAccount);
18   };
19
20   return (
21     <div>
22       <form>
23         <h1>Đăng nhập</h1>
24         <div className="form-group">...
25         <div className="form-group">...
26       </form>
27
28       <button onClick={setDefaultValue}>Set default value</button>
29     </div>
30   );
31 };
32
33 export default SignInScreen;
```