**Overview of React Form Validation using Hooks example**

We will implement validation for a React Form using [React Hook Form](https://react-hook-form.com/) 7 and Bootstrap 4. The form has:

* Full Name: required
* Username: required, from 6 to 20 characters
* Email: required, email format
* Password: required, from 6 to 40 characters
* Confirm Password: required, same as Password
* Accept Terms Checkbox: required

We’re gonna use following modules:

* react 17/16
* bootstrap 4
* react-hook-form 7
* yup
* @hookform/resolvers 2.4.0

**Setup Project**

First we need to install necessary modules.  
Open your project root folder, run the command:

npm install react-hook-form yup @hookform/resolvers

Open *package.json*, the dependencies will look like this:

{

"name": "react-form-validation-hooks",

"version": "0.1.0",

"private": true,

"dependencies": {

...

"@hookform/resolvers": "^2.4.0",

"react": "^17.0.2",

"react-dom": "^17.0.2",

"react-hook-form": "^7.4.2",

"yup": "^0.32.9"

},

...

}

**Import Bootstrap 4**

Open **public**/*index.html* and add following line to <head> tag:

<!DOCTYPE html>

<html lang="en">

<head>

...

<link href="//netdna.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet" />

...

<title>React Form Validation Hooks</title>

</head>

<body>

...

<div id="root"></div>

</body>

</html>

Or you can run command: npm install bootstrap.

Then open **src**/*App.js* and modify the code inside it as following-

import React from "react";

import "bootstrap/dist/css/bootstrap.min.css";

function App() {

return (

...

);

}

export default App;

**Form Validation with React Hook Form 7**

The app component contains Form Validation example built with the React Hook Form library version 7.

Open **src**/*App.js*, we’re gonna import necessary library first:

import React from 'react';

import { useForm } from 'react-hook-form';

import { yupResolver } from '@hookform/resolvers/yup';

import \* as Yup from 'yup';

Next we use [Yup](https://github.com/jquense/yup) schema validation library for Form validation rules:

function App() {

const validationSchema = Yup.object().shape({

fullname: Yup.string().required('Fullname is required'),

username: Yup.string()

.required('Username is required')

.min(6, 'Username must be at least 6 characters')

.max(20, 'Username must not exceed 20 characters'),

email: Yup.string()

.required('Email is required')

.email('Email is invalid'),

password: Yup.string()

.required('Password is required')

.min(6, 'Password must be at least 6 characters')

.max(40, 'Password must not exceed 40 characters'),

confirmPassword: Yup.string()

.required('Confirm Password is required')

.oneOf([Yup.ref('password'), null], 'Confirm Password does not match'),

acceptTerms: Yup.bool().oneOf([true], 'Accept Terms is required')

});

...

}

Then we pass the rules to the React Hook Form [useForm()](https://react-hook-form.com/api/useform) function using yupResolver() function from [react-hook-form/resolvers](https://github.com/react-hook-form/resolvers) library.

function App() {

const validationSchema = ...

const {

register,

handleSubmit,

reset,

formState: { errors }

} = useForm({

resolver: yupResolver(validationSchema)

});

...

}

The useForm() hook function returns an object that we use following methods:

* register: register inputs
* handleSubmit: handle form submit
* reset:reset the form

The object also has formState that contains errors.

When the form is valid and submitted, onSubmit() method is called and the form data will show in console:

function App() {

...

const onSubmit = data => {

console.log(JSON.stringify(data, null, 2));

};

...

}

**React Hook Form Validation template**

Now we create the form with input fields and validation messages. We register form fields with the React Hook Form by calling the register function above with the field name of input element {...register('fieldname')}.

function App() {

...

return (

<div className="register-form">

<form onSubmit={handleSubmit(onSubmit)}>

<div className="form-group">

<label>Full Name</label>

<input

name="fullname"

type="text"

{...register('fullname')}

className={`form-control ${errors.fullname ? 'is-invalid' : ''}`}

/>

<div className="invalid-feedback">{errors.fullname?.message}</div>

</div>

<div className="form-group">

<label>Username</label>

<input

name="username"

type="text"

{...register('username')}

className={`form-control ${errors.username ? 'is-invalid' : ''}`}

/>

<div className="invalid-feedback">{errors.username?.message}</div>

</div>

<div className="form-group">

<label>Email</label>

<input

name="email"

type="text"

{...register('email')}

className={`form-control ${errors.email ? 'is-invalid' : ''}`}

/>

<div className="invalid-feedback">{errors.email?.message}</div>

</div>

<div className="form-group">

<label>Password</label>

<input

name="password"

type="password"

{...register('password')}

className={`form-control ${errors.password ? 'is-invalid' : ''}`}

/>

<div className="invalid-feedback">{errors.password?.message}</div>

</div>

<div className="form-group">

<label>Confirm Password</label>

<input

name="confirmPassword"

type="password"

{...register('confirmPassword')}

className={`form-control ${

errors.confirmPassword ? 'is-invalid' : ''

}`}

/>

<div className="invalid-feedback">

{errors.confirmPassword?.message}

</div>

</div>

<div className="form-group form-check">

<input

name="acceptTerms"

type="checkbox"

{...register('acceptTerms')}

className={`form-check-input ${

errors.acceptTerms ? 'is-invalid' : ''

}`}

/>

<label htmlFor="acceptTerms" className="form-check-label">

I have read and agree to the Terms

</label>

<div className="invalid-feedback">{errors.acceptTerms?.message}</div>

</div>

<div className="form-group">

<button type="submit" className="btn btn-primary">

Register

</button>

<button

type="button"

onClick={reset}

className="btn btn-warning float-right"

>

Reset

</button>

</div>

</form>

</div>

);

};

export default App;

**Run React Form Validation Hooks App**

You can run our App with command: npm start.  
If the process is successful, open Browser with Url: http://localhost:3000/ and check it.