



React Form Custom Validation with Error Message Example

Search ...

Last updated on: February 13, 2022 | By: [Jolly.exe](#)

FEATURED VIDEOS

Powered by [\[primis\]](#)

NOW PLAYING



..... custom validation and error message example application.

ADVERTISEMEN

RECENT POSTS

Angular 13 How to Manage Search Call using RxJS Debounce ?

Phone (Mobile) Validation Using ReGex in React StackBlitz Example

Angular Material 12 Simple Table Pagination Example

Angular 12 Material Design Example – Positions, Layouts, Events 1 Tutorial

[Picker Calendar Component Example](#)

[Custom Email Validation Pattern in React Js](#)

[Chrome Styled Color-React Js Application](#)

[Ionic 5 Range Slider Example Single, Multiple Markers with Custom Styling](#)

## React Form with Validation

Name :  
  
Name field is required

Email :  
  
Email Field is Invalid

Password :  
  
Password field is required



```
[{ name : 'freaky', email : 'contact@freakyjolly.com', password : 'freaky', "nameError":"","emailError":"","passwordError":""}]
```

ADVERTISEMENT

## How to Add Form with Custom Validation in React App?

On the way, follow these quick steps to set up a form with custom validation in react js application:

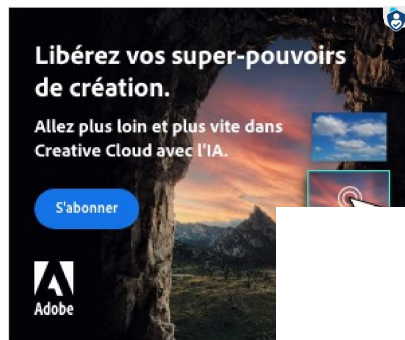
1. Create React Application
2. Install Bootstrap Library
3. Create Form Component with Validation Pattern
4. Using Form Component in App.js
5. Run React App

CATEGORIES

Select Category



Start by creating a new React app, head towards the terminal window and execute following command:



```
npx create-react-app react-demo-app
```

Move into the application directory:

```
cd react-demo-app
```

## Install Bootstrap Library

Next, we install the Bootstrap library package to quickly style our app. Execute the following npm command to install the bootstrap package:

```
npm install bootstrap --save
```

**formvalidation.component.js.** This will have our form component with validation. It will validate the form controls defined and show errors to validate before submitting the values to the parent.

Open the **src/formvalidation.component.js** file and update it with below code:

```
import React from "react";
const defaultState = {
  name: '',
  email: '',
  password: '',
  nameError: '',
  emailError: '',
  passwordError: '',
};
class FormValidationComponent extends React.Component {
  constructor() {
    super();
    this.state = defaultState;
    this.handleChange = this.handleChange.bind(this);
  }
}
```

Add reCAPTCHA v2 as Validation using Cust

Angular 8/7 Form Val with Bootstrap using driven Validation

Formik React Forms t Bootstrap with Valida Tutorial with Example

Angular 8 | Template Form Validation Tutor Example

React DatePicker and Timepicker Compon Validation using react datepicker Example A

Ionic React | Introduc Latest React beta ver Ionic 4

Angular 9|8 Radio Lis Validation using Reac

Setup Google Analyti React Application - R Tutorial by Example

AutoResizable Textare in React as User Type react-textarea-autosiz



ezoic

AngularIonic FrameworkReactAngular MaterialFirebaseAndroid

```
const target = event.target;
var value = target.value;
const name = target.name;
this.setState({
  [name]: value,
});
}
validate() {
  let nameError = "";
  let emailError = "";
  let passwordError = "";
  if (!this.state.name) {
    nameError = "Name field is required";
  }
  const reg = /^w+([\.-]?\w+)*@w+([\.-]?\w+)*(\.w{2,3})+$/;
  if (!this.state.email || reg.test(this.state.email) === false) {
    emailError = "Email Field is Invalid ";
  }
  if (!this.state.password) {
    passwordError = "Password field is required";
  }
  if (emailError || nameError || passwordError) {
    this.setState({ nameError, emailError, passwordError });
    return false;
  }
  return true;
}
submit() {
  if (this.validate()) {

    //Send back the form values to the parent i.e App.js
    this.props.onSubmit([this.state]);

    this.setState(defaultState);
  }
}
```

ADVERTISEMEN

1

```
<div className="row">
  <div className="col-md-6 offset-md-3">
    <h3>React Custom Form Validation Example - FreakyJolly.com</h3>
    <br />
    <div className="form-row">
      <div className="form-group col-md-6">
        <label>Name :</label>
        <input
          type="text"
          className="form-control"
          name="name"
          value={this.state.name}
          onChange={this.handleInputChange}
        />
        <span className="text-danger">{this.state.nameError}</span>
      </div>
    </div>
    <div className="form-row">
      <div className="form-group col-md-6">
        <label>Email :</label>
        <input
          type="email"
          className="form-control"
          name="email"
        />
      </div>
    </div>
  </div>
```

[Angular](#)[Ionic Framework](#)[React](#)[Angular Material](#)[Firebase](#)[Android](#)

```

    />
    <span className="text-danger">{this.state.emailError}</span>
  </div>
</div>
<div className="form-row">
  <div className="form-group col-md-6">
    <label>Password :</label>
    <input
      type="password"
      className="form-control"
      name="password"
      value={this.state.password}
      onChange={this.handleInputChange}
    />
    <span className="text-danger">{this.state.passwordError}</span>
  </div>
</div>
<div className="form-row">
  <div className="col-md-12 text-center">
    <button
      type="submit"
      className="btn btn-primary"
      onClick={() => this.submit()}
    >
      Submit
    </button>
  </div>
</div>
</div>
</div>
</div>
  );
}
}
export default FormValidationComponent;

```



## Using Form Component in App.js

Now, we will import our FormValidationComponent into the App.js file. In the App component, we also define a `onFormSubmit` the event handler, that will be passed to our compoennt to emit the form values when submitted.

In addition, we also have used the **bootstrap.min.css** file to style our form with bootstrap classes.



```
import { Component } from "react";
import "../App.css";
import "../node_modules/bootstrap/dist/css/bootstrap.min.css";
import FormValidationComponent from "../formvalidation.component";

class App extends Component {

  constructor() {
    super();
    this.state = {
      formValues:''
    };
  }

  onFormSubmit = event => {
    // Get Form Values
    console.log(event);
    this.setState({formValues:JSON.stringify(event)})
  };

  render() {
    return (
      <div className="App container">
        <FormValidationComponent onSubmit={this.onFormSubmit}/>

        <div>
          <h5>Submitted values</h5>
          <code>{this.state.formValues}</code>
        </div>
      </div>
    );
  }
}

export default App;
```

## Run React App

Finally, run the application by hitting the below command:

```
npm start
```

it will open the application at the following URL:

```
http://localhost:3000
```

## Conclusion

We discussed how to add a custom form in React js application. Also, implemented the custom validation with error messages to inform the user if fields are invalid. In addition to that, we emitted the form values back to the App.js component.