

30 Days Of React: Form Using React Hooks



LinkedIn



Follow @asabeneh

776

Author: [Asabeneh Yetayeh](#)

October, 2020

<< [Day 21](#) | [Day 23](#)>>

- [Introducing React Hook](#)
 - [Basic Hooks](#)
 - [State Hook](#)
 - [Effect Hook](#)
 - [Context Hook](#)
 - [Additional Hook](#)
- [Exercises](#)
 - [Exercises: Level 1](#)

Forms

Form is used to collect data from a user. Once in a while we use form to fill our information on a paper or on a website. Either to sign up, sign in or to apply for a job, we fill different form fields to submit our data to remote database. We encounter different form fields when we fill a form such as simple text, email, password, telephone, date, checkbox, radio button, option selection and text area field. Currently, HTML5 has provide quite a lot of field types. You may have a look at the following available HTML5 input types.

```
<input type="text" />
<input type="number" />
```

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

<input type="email" />
<input type="password" />
<input type="tel" />

<input type="checkbox" />
<input type="radio" />

<input type="color" />

<input type="url" />
<input type="image" />
<input type="file" />

<input type="hidden" />

<input type="date" />
<input type="datetime-local" />
<input type="month" />
<input type="week" />
<input type="time" />

<input type="reset" />
<input type="search" />
<input type="submit" />
<input type="button" />
```

Another HTML fields to get data from a form are textarea and select with options elements.

```
<textarea>Please write your comment ...</textarea>

<select name="country">
  <option value="">Select your country</option>
  <option value="finland">Finland</option>
  <option value="sweden">Sweden</option>
  <option value="denmark">Denmark</option>
  <option value="norway">Norway</option>
  <option value="iceland">Iceland</option>
</select>
```

Now, you know most of the fields we need to get data from a form. Let's start with an input with type text field. In the previous lessons, we saw different types of events and today we will focus on more of *onChange* event type which triggers whenever an input field data changes. Input field has by default a memory to store input data but in this section we control that using state and we implement a controlled input. Today we will implement a controlled input.

Getting data from an input field

So far used class based components to use state and to get data from controlled input but in this section we will use useState hooks. Now, it is time to learn how to get data from an input field using hooks. We need an input field, event listener (onChange) and state to get data from a controlled input. See the example below. The h1 element below the input tag display what we write on the input. Check live [demo](#).

The input element has many attributes such as value, name, id, placeholder, type and event handler. In addition, we can link a label and an input field using an id of input field and htmlFor of the label. If label and input are linked it will focus the input when we click on label. Look at the example give below.

```
import React, { useState } from 'react'
import ReactDOM from 'react-dom'

const App = (props) => {
  // initial state and method to update state
  const [firstName, setFirstName] = useState('')
  const handleChange = (e) => {
    const value = e.target.value
    setFirstName(value)
  }
  return (
    <div className='App'>
      <label htmlFor='firstName'>First Name: </label>
      <input
        type='text'
        id='firstName'
        name='firstName'
        placeholder='First Name'
        value={firstName}
        onChange={handleChange}
      />
      <h1>{firstName}</h1>
    </div>
  )
}

const rootElement = document.getElementById('root')
ReactDOM.render(<App />, rootElement)
```

We usually use form to handle user information. Let us move to form section and make use the form element.

Getting multiple input data from form

In this section we will develop a small form which collect user information. Our user is a student. We use a parent form element and certain number of input elements to collect user information. In addition to that we will have event listener for the form (onSubmit) and for the inputs (onChange). See the following example try to see the comments too. You can also check the live [demo](#).

As you can see we have four fields, if you we create a separate method to update all the fields we will have method for updating(firstName, lastName, country and title) instead let's have one method which can update all.

```
import React, { useState } from 'react'
import ReactDOM from 'react-dom'

const App = (props) => {
  const initialState = {
    firstName: '',
    lastName: '',
    country: '',
    title: '',
  }
  const [formData, setData] = useState(initialState)

  const onChange = (e) => {
    const { name, value } = e.target
    setData({ ...formData, [name]: value })
  }

  const onSubmit = (e) => {
    /*
     e.preventDefault()
     stops the default behavior of form element
     specifically refreshing of page
    */
    e.preventDefault()

    /*
     the is the place where we connect backend api
     to send the data to the database
    */
    console.log(formData)
  }

  // accessing the state value by destrutcturing the state
  const { firstName, lastName, title, country } = formData
  return (
    <div className='App'>
      <h3>Add Student</h3>
      <form onSubmit={onSubmit}>
        <div>
          <input
            type='text'
            name='firstName'
            placeholder='First Name'
            value={firstName}
            onChange={onChange}
          />
        </div>
        <div>
          <input
            type='text'
            name='lastName'
            placeholder='Last Name'
            value={lastName}
          />
        </div>
      </form>
    </div>
  )
}
```

```

        onChange={onChange}
      />
    </div>
    <div>
      <input
        type='text'
        name='country'
        placeholder='Country'
        value={country}
        onChange={onChange}
      />
    </div>
    <div>
      <input
        type='text'
        name='title'
        placeholder='Title'
        value={title}
        onChange={onChange}
      />
    </div>

    <button class='btn btn-success'>Submit</button>
  </form>
</div>
)
}

const rootElement = document.getElementById('root')
ReactDOM.render(<App />, rootElement)

```

The above form handles only text types but do have different input field types. Let's do another form which handle all the different input field types.

Get data from different input field types

```

// index.js
import React, { useState } from 'react'
import ReactDOM from 'react-dom'

const options = [
  {
    value: '',
    label: '-- Select Country--',
  },
  {
    value: 'Finland',
    label: 'Finland',
  },
  {
    value: 'Sweden',

```

```

    label: 'Sweden',
  },
  {
    value: 'Norway',
    label: 'Norway',
  },
  {
    value: 'Denmark',
    label: 'Denmark',
  },
]

// mapping the options to list(array) of JSX options

const selectOptions = options.map(({ value, label }) => (
  <option key={label} value={value}>
    {' '}
    {label}
  </option>
))

const App = (props) => {
  const initialState = {
    firstName: '',
    lastName: '',
    email: '',
    title: '',
    country: '',
    tel: '',
    dateOfBirth: '',
    favoriteColor: '',
    weight: '',
    gender: '',
    file: '',
    bio: '',
    skills: {
      html: false,
      css: false,
      javascript: false,
    },
  }
  const [formData, setFormData] = useState(initialState)

  const onChange = (e) => {
    /*
     we can get the name and value like: e.target.name, e.target.value
     Wwe can also destructure name and value from e.target
     const name = e.target.name
     const value = e.target.value
     */
    const { name, value, type, checked } = e.target
    /*
     [variablename] we can make a value stored in a certain variable could be a key
     for an object, in this case a key for the state

```

```
*/

if (type === 'checkbox') {
  setFormData({
    ...formData,
    skills: { ...formData.skills, [name]: checked },
  })
} else if (type === 'file') {
  setFormData({ ...formData, [name]: e.target.files[0] })
} else {
  setFormData({ ...formData, [name]: value })
}
}

const onSubmit = (e) => {
  /*
  e.preventDefault()
  stops the default behavior of form element
  specifically refreshing of page
  */
  e.preventDefault()
  const {
    firstName,
    lastName,
    title,
    email,
    tel,
    dateOfBirth,
    favoriteColor,
    weight,
    country,
    gender,
    bio,
    file,
    skills,
  } = formData

  const formattedSkills = []
  for (const key in skills) {
    console.log(key)
    if (skills[key]) {
      formattedSkills.push(key.toUpperCase())
    }
  }
  const data = {
    firstName,
    lastName,
    title,
    email,
    tel,
    dateOfBirth,
    favoriteColor,
    weight,
    country,
    gender,
```

```
    bio,
    file,
    skills: formattedSkills,
  }
  /*
  the is the place where we connect backend api
  to send the data to the database
  */
  console.log(data)
}

// accessing the state value by destrutcturing the state
const {
  firstName,
  lastName,
  title,
  country,
  email,
  tel,
  dateOfBirth,
  favoriteColor,
  weight,
  gender,
  bio,
} = formData
return (
  <div className='App'>
    <h3>Add Student</h3>
    <form onSubmit={onSubmit}>
      <div className='row'>
        <div className='form-group'>
          <label htmlFor='firstName'>First Name </label>
          <input
            type='text'
            id='firstName'
            name='firstName'
            value={firstName}
            onChange={onChange}
            placeholder='First Name'
          />
        </div>
        <div className='form-group'>
          <label htmlFor='lastName'>Last Name </label>
          <input
            type='text'
            id='lastName'
            name='lastName'
            value={lastName}
            onChange={onChange}
            placeholder='Last Name'
          />
        </div>
        <div className='form-group'>
          <label htmlFor='title'>Title </label>
```



```
        <input
          type='text'
          id='title'
          name='title'
          placeholder='Title'
          value={title}
          onChange={onChange}
        />
      </div>
      <div className='form-group'>
        <label htmlFor='email'>Email </label>
        <input
          type='email'
          id='email'
          name='email'
          value={email}
          onChange={onChange}
          placeholder='Email'
        />
      </div>
    </div>

    <div className='form-group'>
      <label htmlFor='tel'>Telephone </label>
      <input
        type='tel'
        id='tel'
        name='tel'
        value={tel}
        onChange={onChange}
        placeholder='Tel'
      />
    </div>

    <div className='form-group'>
      <label htmlFor='dateOfBirth'>Date of birth </label>
      <input
        type='date'
        id='dateOfBirth'
        name='dateOfBirth'
        value={dateOfBirth}
        onChange={onChange}
        placeholder='Date of Birth'
      />
    </div>
    <div className='form-group'>
      <label htmlFor='favoriteColor'>Favorite Color</label>
      <input
        type='color'
        id='color'
        name='favoriteColor'
        value={favoriteColor}
        onChange={onChange}
        placeholder='Favorite Color'
      />
    </div>
```

```
    />
  </div>
  <div className='form-group'>
    <label htmlFor='weight'>Weight </label>
    <input
      type='number'
      id='weight'
      name='weight'
      value={weight}
      onChange={onChange}
      placeholder='Weight in Kg'
    />
  </div>
  <div>
    <label htmlFor='country'>Country</label> <br />
    <select
      name='country'
      onChange={onChange}
      id='country'
      value={country}
    >
      {selectOptions}
    </select>
  </div>

  <div>
    <p>Gender</p>
    <div>
      <input
        type='radio'
        id='female'
        name='gender'
        value='Female'
        onChange={onChange}
        checked={gender === 'Female'}
      />
      <label htmlFor='female'>Female</label>
    </div>
    <div>
      <input
        id='male'
        type='radio'
        name='gender'
        value='Male'
        onChange={onChange}
        checked={gender === 'Male'}
      />
      <label htmlFor='male'>Male</label>
    </div>
    <div>
      <input
        id='other'
        type='radio'
        name='gender'
```

```

        value='Other'
        onChange={onChange}
        checked={gender === 'Other'}
      />
      <label htmlFor='other'>Other</label>
    </div>
  </div>

  <div>
    <p>Select your skills</p>
    <div>
      <input type='checkbox' id='html' name='html' onChange={onChange} />
      <label htmlFor='html'>HTML</label>
    </div>
    <div>
      <input type='checkbox' id='css' name='css' onChange={onChange} />
      <label htmlFor='css'>CSS</label>
    </div>
    <div>
      <input
        type='checkbox'
        id='javascript'
        name='javascript'
        onChange={onChange}
      />
      <label htmlFor='javascript'>JavaScript</label>
    </div>
  </div>
  <div>
    <label htmlFor='bio'>Bio</label> <br />
    <textarea
      id='bio'
      name='bio'
      value={bio}
      onChange={onChange}
      cols='120'
      rows='10'
      placeholder='Write about yourself ...'
    />
  </div>

  <div>
    <input type='file' name='file' onChange={onChange} />
  </div>
  <div>
    <button>Submit</button>
  </div>
</form>
</div>
)
}

const rootElement = document.getElementById('root')
ReactDOM.render(<App />, rootElement)

```

Form Validation

What is validation?

The action or process of checking or proving the validity or accuracy of something in this case data.

What is the purpose of validation

The main purpose to validation is to get a desired data from users. In addition, to prevent malicious users and data.

Validation Types

Validation can be done in client side or sever side. At the moment, we are using React which is a front end technology and we use client side validation.A validation can implement using HTML5 built-in validation or using JavaScript(using regular expression).

In the following snippet of code, a validation has been implemented the first field. Try to understand how it works. The onBlur event has been used to check validity when the input is not focused.

```
// index.js
import React, { useState } from 'react'
import ReactDOM from 'react-dom'

const options = [
  {
    value: '',
    label: '-- Select Country--',
  },
  {
    value: 'Finland',
    label: 'Finland',
  },
  {
    value: 'Sweden',
    label: 'Sweden',
  },
  {
    value: 'Norway',
    label: 'Norway',
  },
  {
    value: 'Denmark',
    label: 'Denmark',
  },
]

// mapping the options to list(array) of JSX options

const selectOptions = options.map(({ value, label }) => (
```

```

    <option key={label} value={value}>
      {' '}
      {label}
    </option>
  ))

const App = (props) => {
  const initialState = {
    firstName: '',
    lastName: '',
    email: '',
    title: '',
    country: '',
    tel: '',
    dateOfBirth: '',
    favoriteColor: '',
    weight: '',
    gender: '',
    file: '',
    bio: '',
    skills: {
      html: false,
      css: false,
      javascript: false,
    },
    touched: {
      firstName: false,
      lastName: false,
    },
  }
  const [formData, setFormData] = useState(initialState)

  const onChange = (e) => {
    /*
     we can get the name and value like: e.target.name, e.target.value
     Wwe can also destructure name and value from e.target
     const name = e.target.name
     const value = e.target.value
     */
    const { name, value, type, checked } = e.target
    /*
    [variablename] we can make a value stored in a certain variable could be a key
    for an object, in this case a key for the state
    */

    if (type === 'checkbox') {
      setFormData({
        ...formData,
        skills: { ...formData.skills, [name]: checked },
      })
    } else if (type === 'file') {
      setFormData({ ...formData, [name]: e.target.files[0] })
    } else {
      setFormData({ ...formData, [name]: value })
    }
  }
}

```

```
    }  
  }  
  const onSubmit = (e) => {  
    /*  
    e.preventDefault()  
    stops the default behavior of form element  
    specifically refreshing of page  
    */  
    e.preventDefault()  
    const {  
      firstName,  
      lastName,  
      title,  
      email,  
      tel,  
      dateOfBirth,  
      favoriteColor,  
      weight,  
      country,  
      gender,  
      bio,  
      file,  
      skills,  
    } = formData  
  
    const formattedSkills = []  
    for (const key in skills) {  
      console.log(key)  
      if (skills[key]) {  
        formattedSkills.push(key.toUpperCase())  
      }  
    }  
    const data = {  
      firstName,  
      lastName,  
      title,  
      email,  
      tel,  
      dateOfBirth,  
      favoriteColor,  
      weight,  
      country,  
      gender,  
      bio,  
      file,  
      skills: formattedSkills,  
    }  
    /*  
    the is the place where we connect backend api  
    to send the data to the database  
    */  
    console.log(data)  
  }  
  const onBlur = (e) => {
```

```

    const { name } = e.target
    setFormData({ ...formData, touched: { ...formData.touched, [name]: true } })
  }
  const validate = () => {
    // Object to collect error feedback and to display on the form
    const errors = {
      firstName: '',
    }

    if (
      (formData.touched.firstName && formData.firstName.length < 3) ||
      (formData.touched.firstName && formData.firstName.length > 12)
    ) {
      errors.firstName = 'First name must be between 2 and 12'
    }
    return errors
  }

  // accessing the state value by destructuring the state
  const {
    firstName,
    lastName,
    title,
    country,
    email,
    tel,
    dateOfBirth,
    favoriteColor,
    weight,
    gender,
    bio,
  } = formData

  const errors = validate()

  return (
    <div className='App'>
      <h3>Add Student</h3>
      <form onSubmit={onSubmit}>
        <div className='row'>
          <div className='form-group'>
            <label htmlFor='firstName'>First Name </label>
            <input
              type='text'
              id='firstName'
              name='firstName'
              value={firstName}
              onChange={onChange}
              onBlur={onBlur}
              placeholder='First Name'
            />
            <br />
            {errors.firstName && <small>{errors.firstName}</small>}
          </div>

```

```
<div className='form-group'>
  <label htmlFor='lastName'>Last Name </label>
  <input
    type='text'
    id='lastName'
    name='lastName'
    value={lastName}
    onChange={onChange}
    placeholder='Last Name'
  />
</div>
<div className='form-group'>
  <label htmlFor='title'>Title </label>
  <input
    type='text'
    id='title'
    name='title'
    placeholder='Title'
    value={title}
    onChange={onChange}
  />
</div>
<div className='form-group'>
  <label htmlFor='email'>Email </label>
  <input
    type='email'
    id='email'
    name='email'
    value={email}
    onChange={onChange}
    placeholder='Email'
  />
</div>
</div>

<div className='form-group'>
  <label htmlFor='tel'>Telephone </label>
  <input
    type='tel'
    id='tel'
    name='tel'
    value={tel}
    onChange={onChange}
    placeholder='Tel'
  />
</div>

<div className='form-group'>
  <label htmlFor='dateOfBirth'>Date of birth </label>
  <input
    type='date'
    id='dateOfBirth'
    name='dateOfBirth'
    value={dateOfBirth}
```



```

        onChange={onChange}
        placeholder='Date of Birth'
      />
    </div>
    <div className='form-group'>
      <label htmlFor='favoriteColor'>Favorite Color</label>
      <input
        type='color'
        id='color'
        name='favoriteColor'
        value={favoriteColor}
        onChange={onChange}
        placeholder='Favorite Color'
      />
    </div>
    <div className='form-group'>
      <label htmlFor='weight'>Weight </label>
      <input
        type='number'
        id='weight'
        name='weight'
        value={weight}
        onChange={onChange}
        placeholder='Weight in Kg'
      />
    </div>
    <div>
      <label htmlFor='country'>Country</label> <br />
      <select
        name='country'
        onChange={onChange}
        id='country'
        value={country}
      >
        {selectOptions}
      </select>
    </div>

    <div>
      <p>Gender</p>
      <div>
        <input
          type='radio'
          id='female'
          name='gender'
          value='Female'
          onChange={onChange}
          checked={gender === 'Female'}
        />
        <label htmlFor='female'>Female</label>
      </div>
      <div>
        <input
          id='male'

```

```
        type='radio'
        name='gender'
        value='Male'
        onChange={onChange}
        checked={gender === 'Male'}
      />
      <label htmlFor='male'>Male</label>
    </div>
    <div>
      <input
        id='other'
        type='radio'
        name='gender'
        value='Other'
        onChange={onChange}
        checked={gender === 'Other'}
      />
      <label htmlFor='other'>Other</label>
    </div>
  </div>

  <div>
    <p>Select your skills</p>
    <div>
      <input type='checkbox' id='html' name='html' onChange={onChange} />
      <label htmlFor='html'>HTML</label>
    </div>
    <div>
      <input type='checkbox' id='css' name='css' onChange={onChange} />
      <label htmlFor='css'>CSS</label>
    </div>
    <div>
      <input
        type='checkbox'
        id='javascript'
        name='javascript'
        onChange={onChange}
      />
      <label htmlFor='javascript'>JavaScript</label>
    </div>
  </div>
  <div>
    <label htmlFor='bio'>Bio</label> <br />
    <textarea
      id='bio'
      name='bio'
      value={bio}
      onChange={onChange}
      cols='120'
      rows='10'
      placeholder='Write about yourself ...'
    />
  </div>
```

```
    <div>
      <input type='file' name='file' onChange={onChange} />
    </div>
    <div>
      <button>Submit</button>
    </div>
  </form>
</div>
)
}

const rootElement = document.getElementById('root')
ReactDOM.render(<App />, rootElement)
```

Exercises

Exercises: Level 1

1. What is the importance of form?
2. How many input types do you know?
3. Mention at least four attributes of an input element
4. What is the importance of htmlFor?
5. Write an input type which is not given in the example if there is?
6. What is a controlled input?
7. What do you need to write a controlled input?
8. What event type do you use to listen changes on an input field?
9. What is the value of a checked checkbox?
10. When do you use onChange, onBlur, onSubmit?
11. What is the purpose of writing e.preventDefault() inside the submit handler method?
12. How do you bind data in React? The first input field example is data binding in React.
13. What is validation?
14. What is the event type you use to listen when an input changes?
15. What are event types do you use to validate an input?

Exercises: Level 2

1. Validate the form given above (a gif image or a video will be provided later). First try to validate without using any library then try it with [validator.js](#).

Exercises: Level 3

Coming ..

🎉 CONGRATULATIONS ! 🎉

[<< Day 21](#) | [Day 23>>](#)