Creating Reusable Custom Form Elements



Nitin Singh Full-stack Developer

Overview



Understand the need for creating custom form fields

Understand how the useField hook provided by Formik works

Create customized input fields

Create non-standard input fields



Need for Custom Form Elements

Reusable components Write less code 2

Consistent UX across app

Change at one place to update all forms on the app

3

Non-standard input components

Create components like star-based reviews

useField React hook provided by Formik is used to create custom <Field> input components



useField Hook



Accepts a props object or a field name string as input

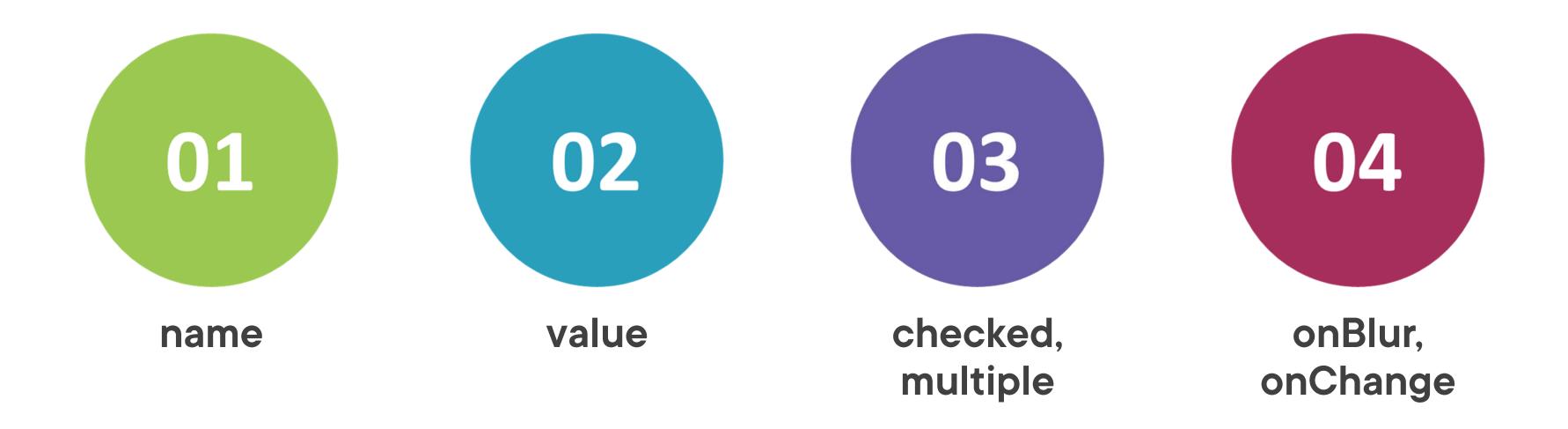


The props object contains the props that you want to pass to the custom Field along with any extra props like label for the Field

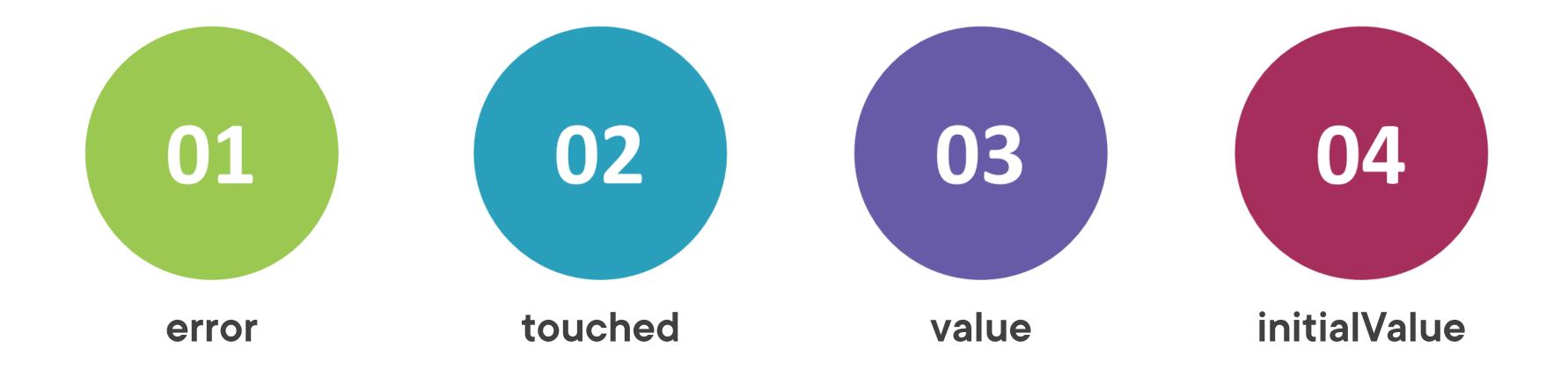


Return an array with 3 elements containing FieldInputProps, FieldMetaProps and FieldHelperProps

FieldInputProps

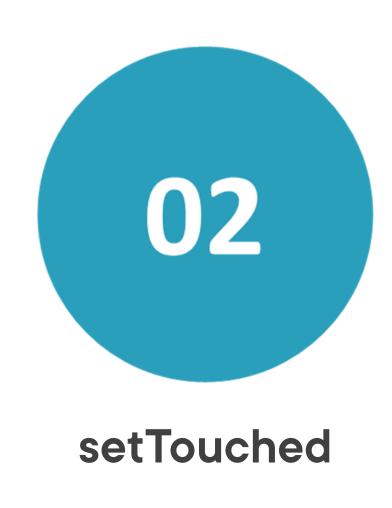


FieldMetaProps



FieldHelperProps







```
const MyTextInput = ({ label, ...props }) => {
   const [field, meta] = useField(props);
   return (
     <>
      <label htmlFor={props.name}>{label}</label>
       <input {...field} {...props} />
       {meta.touched && meta.error ? (
           <div>{meta.error}</div>
        : null}
     </>
   );};
<MyTextInput
    label="First Name"
   name="firstName"
   type="text"
   placeholder="Jane"
/>
```

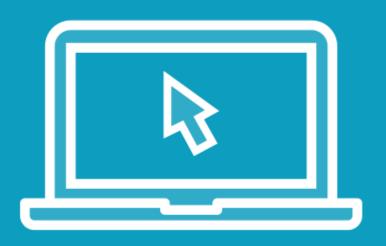
- **◄ Custom input text component**
- Get the field and meta props
- Add label for the input
- Add input field with all field and extra props passed to it
- ◆ Check meta props to decide and show error if needed

■ This is how you would use this custom Field component

```
function RatingComponent(props) {
   const [meta, helpers] = useField(props.name);
   const { value } = meta;
   const { setValue } = helpers;
   const isSelected = v => (v===value ? 'select':'');
   return (
      <div>
       <button onClick={() => setValue(1)
        className={isSelected(1)}>
       </button>
       <button onClick={() => setValue(2)
        className={isSelected(2)}>
       </button>
       <button onClick={() => setValue(3)}
        className={isSelected(3)>
         3
       </button>
      </div>
);}
```

- **◄** A custom rating component
- Get the meta and helper props.
- **◄** Get value from meta props
- Get setValue method from helper props
- **◄** Determine current rating from value
- ◆ On click of button change value using setValue to update the field

Demo



Invoice tracker application

Add Invoice creation form

Create a custom input Field for Email and use it in the invoice creation screen

Create a non-standard input Field for rating a project in the invoice creation screen

Summary



- Benefits of creating custom form fields
- How the useField hook provided by Formik works
- How to create and use custom input fields
- How to create and use non-standard input fields



Up Next: Implementing Uncontrolled Forms Using Vanilla React