

Forms revisited

Section 7

In this section

- React hook form
- Why use a package?
- Reusable inputs
- Place autocomplete
- Leaflet for map view



Why use a package?



Without package

```
import { useState } from 'react';

function Form() {
  const [name, setName] = useState('');
  const [submitted, setSubmitted] = useState(false);
  const [touched, setTouched] = useState(false);

  const isDirty = touched && name !== '';
  const isValid = name.trim() !== '';

  const handleSubmit = () => {
    setSubmitted(true);
    if (isValid) {
      alert(`Submitted name: ${name}`);
    }
  };

  return (
    <form onSubmit={handleSubmit}>
      <input
        value={name}
        onChange={(e) => setName(e.target.value)}
        onBlur={() => setTouched(true)}
        placeholder="Name"
      />
      {!isValid && submitted && <p>Name is required.</p>}
      <p>Dirty: {isDirty ? 'Yes' : 'No'}</p>
      <button type="submit">Submit</button>
    </form>
  );
}
```

With package



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

function Form() {
  const {register, handleSubmit, formState: {errors, isDirty }} = useForm();

  const onSubmit = (data: { name: string }) => {
    alert(`Submitted name: ${data.name}`);
  };

  return (
    <form onSubmit={handleSubmit(onSubmit)}>
      <input
        {...register('name', { required: 'Name is required' })}
        placeholder="Name"
      />
      {errors.name && <p>{errors.name.message}</p>}
      <button disabled={!isDirty} type="submit">Submit</button>
    </form>
  );
}
```

The goal of this section

Summary

In this section

- React hook form
- Why use a package?
- Reusable inputs
- Place autocomplete
- Leaflet for map view



Up next