

React Forms & Validations

By

Narasimha Rao T

Microsoft .Net FSD Trainer

Professional Development Trainer

tnrao.trainer@gmail.com

1. Introduction to React Forms

- In React, forms are used to collect user input.
- Unlike plain HTML, React controls form elements via **state**.
- The goal: keep form data in sync with the component's state.

2. Controlled Components

- A **controlled component** is an input element whose value is controlled by React state.
- Example:

```
function ControlledForm() {
  const [name, setName] = React.useState("");

  return (
    <form>
      <input type="text"
        value={name}
        onChange={(e) => setName(e.target.value)} />
      <p>Hello, {name}</p>
    </form>
  );
}
```

3. Form Handling

- Typical steps in React form handling:
 - i. Initialize state for form fields.
 - ii. Bind state to input's `value` attribute.
 - iii. Update state via `onChange` .
 - iv. Handle submit with `onSubmit` .

- Example:

```
function LoginForm() {
  const [email, setEmail] = React.useState("");
  const [password, setPassword] = React.useState("");

  const handleSubmit = (e) => {
    e.preventDefault();
    console.log("Submitted:", { email, password });
  };

  return (
    <form onSubmit={handleSubmit}>
      <input type="email" value={email}
        onChange={(e) => setEmail(e.target.value)} />
      <input type="password" value={password}
        onChange={(e) => setPassword(e.target.value)} />
      <button type="submit">Login</button>
    </form>
  );
}
```

4. Validation Basics

- Validation ensures user input meets requirements (e.g., required fields, format checks).
- Common strategies:
 - Inline validation (real-time as user types).
 - On-submit validation (when form is submitted).
- Example rules:
 - Required field: `input.trim() !== ""`
 - Email format: Use regex or string methods.

5. Displaying Error Messages

- Store errors in state and display them conditionally.

```
function SignupForm() {
  const [email, setEmail] = React.useState("");
  const [error, setError] = React.useState("");

  const handleSubmit = (e) => {
    e.preventDefault();
    if (!email.includes("@")) {
      setError("Invalid email address");
    } else {
      setError("");
      console.log("Submitted:", email);
    }
};
```

```
return (
  <form onSubmit={handleSubmit}>
    <input type="email" value={email}
      onChange={(e) => setEmail(e.target.value)}
    />
    {error && <p style={{color: "red"}}>{error}</p>}
    <button type="submit">Submit</button>
  </form>
);
}
```

Q & A

Narasimha Rao T

tnrao.trainer@gmail.com