Armen Melkumyan • 1st
Technical / Solutions Architect
5mo •

From React technical interviews: Controlled vs. Uncontrolled Components in React

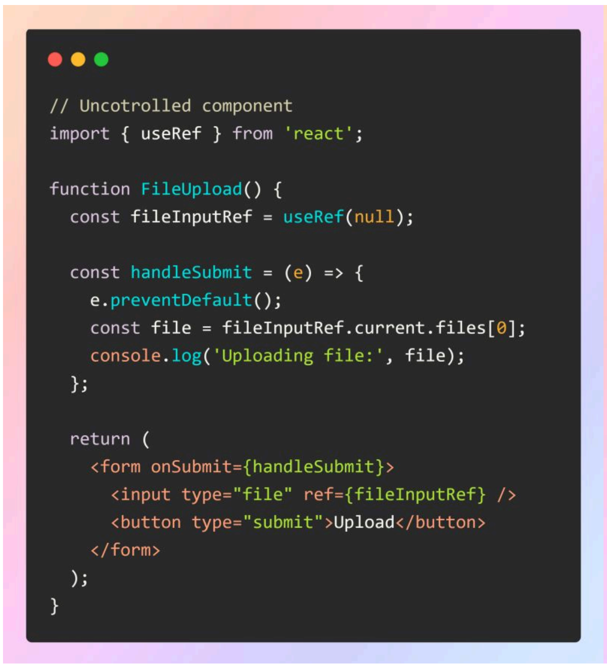
When building forms in React, choosing between controlled and uncontrolled components can impact both your code's flexibility and performance. Let’s dive in with examples!

💡 Controlled Components
In controlled components, React state is the source of truth, keeping the UI and data tightly coupled. This allows for data validation, conditional UI updates, and synchronization across components.
Example: A login form where the username and password are stored in React state, enabling live validation.

💡 Uncontrolled Components
Uncontrolled components are different: refs allow direct access to DOM elements, letting the component manage its own value internally. This approach is ideal when simplicity and performance are priorities, like when handling file uploads.
Example: A file upload form where we retrieve the file data directly from the DOM only when necessary.

Using ref here makes it efficient to access the file data only on submit without tracking it in React state.

⚙️ When to Use Each?
Controlled Components: For forms needing validation, real-time feedback, or tight control over updates.
Uncontrolled Components: When simplicity and lower overhead are beneficial, or you’re integrating with non-React code.
Choosing wisely between these enhances app performance and maintainability.
#React #JavaScript #WebDevelopment #ControlledVsUncontrolled #SeniorDev #TechInterview

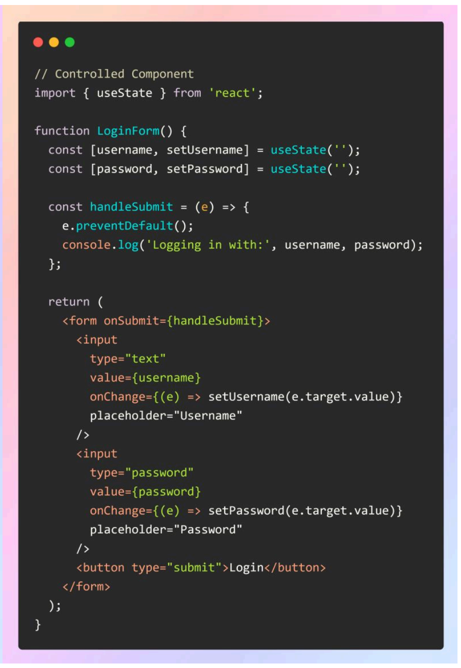


```jsx
// Uncotrolled component
import { useRef } from 'react';

function FileUpload() {
  const fileInputRef = useRef(null);

  const handleSubmit = (e) => {
    e.preventDefault();
    const file = fileInputRef.current.files[0];
    console.log('Uploading file:', file);
  };

  return (
    <form onSubmit={handleSubmit}>
      <input type="file" ref={fileInputRef} />
      <button type="submit">Upload</button>
    </form>
  );
}
```



```jsx
// Controlled Component
import { useState } from 'react';

function LoginForm() {
  const [username, setUsername] = useState('');
  const [password, setPassword] = useState('');

  const handleSubmit = (e) => {
    e.preventDefault();
    console.log('Logging in with:', username, password);
  };

  return (
    <form onSubmit={handleSubmit}>
      <input
        type="text"
        value={username}
        onChange={(e) => setUsername(e.target.value)}
        placeholder="Username"
      />
      <input
        type="password"
        value={password}
        onChange={(e) => setPassword(e.target.value)}
        placeholder="Password"
      />
      <button type="submit">Login</button>
    </form>
  );
}
```

128 2 comments 6 reposts