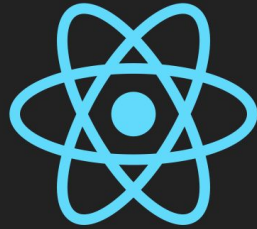


# Lecture 4.1

Handling Forms & User Input



# Topics

- React Forms
  - Form Elements
  - User Input
  - Libraries

# Form Elements

# Form Elements

- React has some **very basic** form support built in.
- This includes **binding values to** the form, binding values **back out** of the form and **accessing form elements**.
- **Built-in support for form validation is not included.**
- React supports the standard selection of form elements, but with some small difference
- HTML form elements work a little bit differently from other DOM elements in React, because form elements naturally keep some internal state.

# HTML Form in React

This form has the default HTML form behavior of browsing to a new page when the user submits the form. If you want this behavior in React, it just works.

But in most cases, it's convenient to have a JavaScript function that handles the submission of the form and has access to the data that the user entered into the form.

```
<form>
  <label>
    Name:
    <input type="text" name="name" />
  </label>
  <input type="submit" value="Submit" />
</form>
```

# Controlled Components

# Controlled Components

- In **HTML**, form elements such as `<input>`, `<textarea>`, and `<select>` typically **maintain their own state** and update it based on user input.
- In **React**, **mutable state** is typically kept in the state property of **components**, and only updated with `setState()`.
- We can make **React state the source of truth**, by having a React component render the form and control the user input in the form.

# Allow User Input

- By default **React form elements** are **read-only**. A simple way to **allow user input** is use React's **component state**.
- To Allow User Input:
  - Add **state** to the component
  - **Bind inputs** to the **component state**
  - Use **onChange** handler to **update state**



```

class NameForm extends React.Component {
  constructor(props) {
    super(props);
    this.state = { value: "" };

    this.handleChange = this.handleChange.bind(this);
  }

  handleChange(event) {
    this.setState({ value: event.target.value });
  }

  render() {
    return (
      <form>
        <label>
          Name:
          <input
            type="text"
            value={this.state.value}
            onChange={this.handleChange}
          />
        </label>
      </form>
    );
  }
}

```

- With a **controlled component**, every state mutation will have an associated handler function.
- This makes it **straightforward** to modify or validate user input.

# Text Input

The **text input** is the same as html element. We need to set the content with the **value prop** in React.

```
<input type="text" value="react" />
```

The **text area** element, is slightly different from html.

In HTML we can provide content in opening and closing tags.

```
<textarea>
|  react
</textarea>
```

In React textarea uses a value prop to set content.

```
<textarea value="react" />
```

# Select Element

The React select element is similar except that the **selected option** is set by providing a **value prop** to the select element.

## HTML Select

```
<select>
  <option value="angular">
    Angular
  </option>
  <option value="react" selected>
    React
  </option>
</select>
```

## React Select

```
<select value="react">
  <option value="angular">
    Angular
  </option>
  <option value="react">
    React
  </option>
</select>
```

# Form Libraries

# Video : React Forms

<https://youtu.be/t3r9xW-sxqs>

# Form Libraries

- Creating forms, making them accessible and implementing good validation can require a lot of effort and maintenance.
- There are many form libraries available for React (as an alternative to control components).
- Using a form library is a compromise. You give up a degree control in exchange for increased productivity and reliability.
- React Json Schema is one library, a simple React component capable of building HTML forms out of JSON schema and using Bootstrap semantics by default.  
<https://mozilla-services.github.io/react-jsonschema-form/>
- React **Formik** is another popular library  
<https://jaredpalmer.com/formik/>