

03-Short-Link-App

Component State and Signup Form

- go to Signup.js
- we are going to create the state object in a constructor function, this gets called when our Signup class first gets instantiated, and it does get called with whatever arguments gets passed in, in the case of our components, jsx passes the props object, then after that we call in the parent constructor calling in super() and passing props, it just makes sure the React component gets the values it needs

```
import React from 'react';
import { Link } from 'react-router';

export default class Signup extends React.Component{

  constructor(props){

    super(props);

    this.sta

  }

  render(){

    return (

      <div>

        <h1>Joint Short Lnk</h1>

        <Link to="/">Already have an account?</Link>

      </div>

    );

  }

}
```

- now we are going to set the state which is just an object

```
import React from 'react';
import { Link } from 'react-router';

export default class Signup extends React.Component{

  constructor(props){

    super(props);
```

```

    this.state = {

    }

}

render(){
    return (
        <div>
            <h1>Joint Short Lnk</h1>

            <Link to="/">Already have an account?</Link>
        </div>
    );
}
}

```

- you can put any value in there, but in our case we are going to put a counter just as an example

```

import React from 'react';
import { Link } from 'react-router';

export default class Signup extends React.Component{
    constructor(props){
        super(props);
        this.state = {
            count: 0,
        }
    }
}

```

- we are now going to render our state in the browser

```

import React from 'react';
import { Link } from 'react-router';

export default class Signup extends React.Component{
    constructor(props){
        super(props);

```

```

    this.state = {
      count: 0,
    }

  }

  render(){
    return (
      <div>
        <h1>Joint Short Lnk</h1>

        <p>{this.state.count}</p>

        <Link to="/">Already have an account?</Link>
      </div>
    );
  }
}

```

- now we are going to figure how to update that state
- in our case lets update our state by adding a button
- when you want to update state this.setState({}) which takes an object, on that object you can provide all of the state properties you want to update

```

import React from 'react';
import { Link } from 'react-router';

export default class Signup extends React.Component{
  constructor(props){
    super(props);
    this.state = {
      count: 0,
    }
  }

  increment(){
    this.setState({
      count : this.state.count + 1
    })
  }
}

```

```

    })
  }
  render(){
    return (
      <div>
        <h1>Joint Short Lnk</h1>

        <p>{this.state.count}</p>

        <button onClick={this.increment.bind(this)}>+1</button>

        <Link to="/">Already have an account?</Link>
      </div>
    );
  }
}

```

- we can also set defaults with our state like here

```

import React from 'react';
import { Link } from 'react-router';

export default class Signup extends React.Component{
  constructor(props){
    super(props);
    this.state = {
      count: this.props.count || 0
    };
  }
  increment(){
    this.setState({
      count : this.state.count + 1
    })
  }
  render(){
    return (
      <div>

```

```

    <h1>Joint Short Lnk</h1>

    <p>{this.state.count}</p>

    <button onClick={this.increment.bind(this)}>+1</button>

    <Link to="/">Already have an account?</Link>

  </div>

  );
}
}

```

- as a challenge we are going to create another button that will decrement
- and we are also going to change the count back to 0

```

import React from 'react';
import { Link } from 'react-router';

export default class Signup extends React.Component{
  constructor(props){
    super(props);
    this.state = {
      count: 0
    };
  }

  increment(){
    this.setState({
      count : this.state.count + 1
    })
  }

  decrement(){
    this.setState({
      count: this.state.count -1
    })
  }

  render(){
    return (

```

```

    <div>
      <h1>Joint Short Lnk</h1>

      <p>{this.state.count}</p>
      <button onClick={this.increment.bind(this)}>+1</button>
      <button onClick={this.decrement.bind(this)}>-1</button>

      <Link to="/">Already have an account?</Link>
    </div>
  );
}
}

```

- now that we have some practice we are going to have an error count instead that keeps track of any error messages. The default error should be an empty string because we don't want it to show on the page until some event happens. And we will also delete the buttons
- first we are going to create a form that will ask the user for a couple of different inputs and button
- and the form will have an onSubmit event that will take an onSubmit method

```

render(){
  return (
    <div>
      <h1>Joint Short Lnk</h1>

      <form onSubmit={this.onSubmit.bind(this)}>
        <input type="email" name="email" placeholder="Email"/>
        <input type="password" name="password" placeholder="Password"/>
        <button>Create Account</button>
      </form>

      <Link to="/">Already have an account?</Link>
    </div>
  );
}
}

```

- we then create that onSubmit method
- and for now we just want to update that error message just to practice a little

```

import React from 'react';
import { Link } from 'react-router';

export default class Signup extends React.Component{
  constructor(props){
    super(props);
    this.state = {
      error: ' '
    };
  }
  onSubmit(e){
    e.preventDefault();
    this.setState({
      error: 'something went wrong'
    })
  }
  render(){
    return (
      <div>
        <h1>Joint Short Lnk</h1>

        <form onSubmit={this.onSubmit.bind(this)}>
          <input type="email" name="email" placeholder="Email"/>
          <input type="password" name="password" placeholder="Password"/>
          <button>Create Account</button>
        </form>

        <Link to="/">Already have an account?</Link>
      </div>
    );
  }
}

```

- we then are going to render that message on the screen
- for this we are going to use the ternary operator

- if there is an error ? we will display the error message but if there is no error : for this case we will have undefined- and that is going to result to render nothing on the screen

```
import React from 'react';
import { Link } from 'react-router';

export default class Signup extends React.Component{
  constructor(props){
    super(props);
    this.state = {
      error: ' '
    };
  }

  onSubmit(e){
    e.preventDefault();
    this.setState({
      error: 'something went wrong'
    })
  }

  render(){
    return (
      <div>
        <h1>Joint Short Lnk</h1>

        {this.state.error ? <p>this.state.error</p>: undefined}

        <form onSubmit={this.onSubmit.bind(this)}>
          <input type="email" name="email" placeholder="Email"/>
          <input type="password" name="password" placeholder="Password"/>
          <button>Create Account</button>
        </form>

        <Link to="/">Already have an account?</Link>
      </div>
    )
  }
}
```



```
);
```

```
}
```

```
}
```

•