Lesson 13

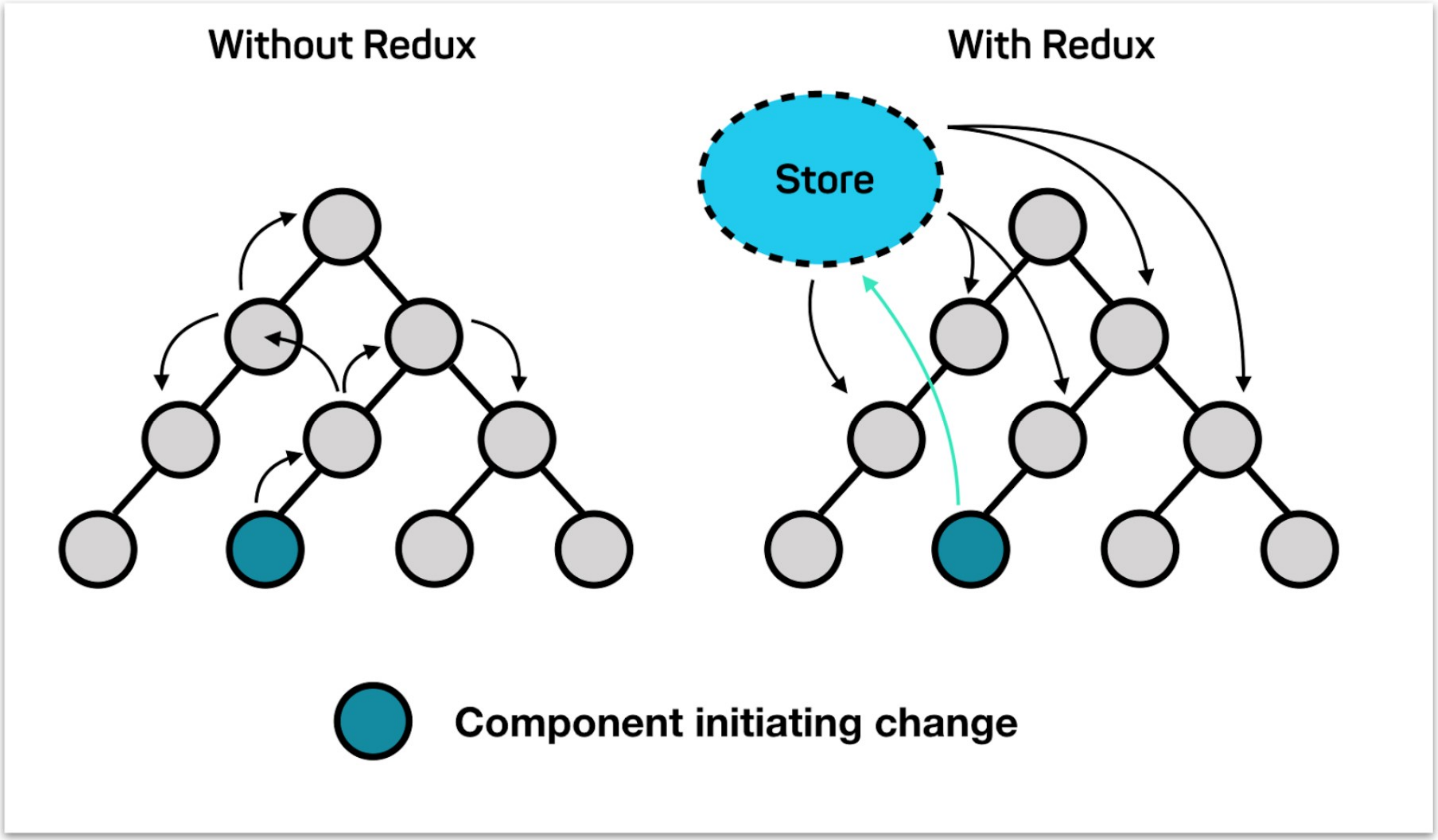
REDUX

What is redux?

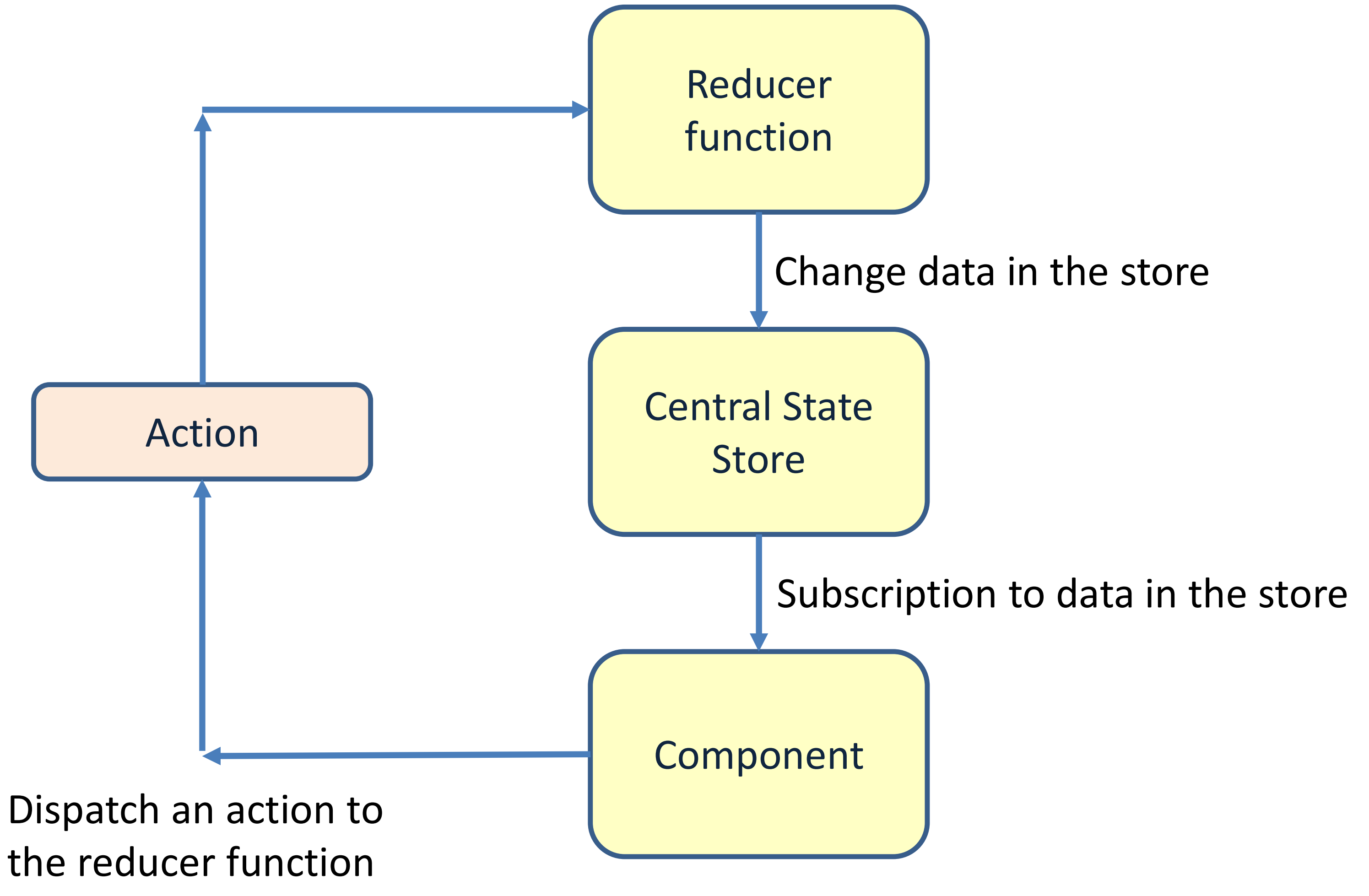
- State management system for app-wide state.

- Local state within 1 component
  - useState()
- State shared between components
  - useState() and props
  - Redux

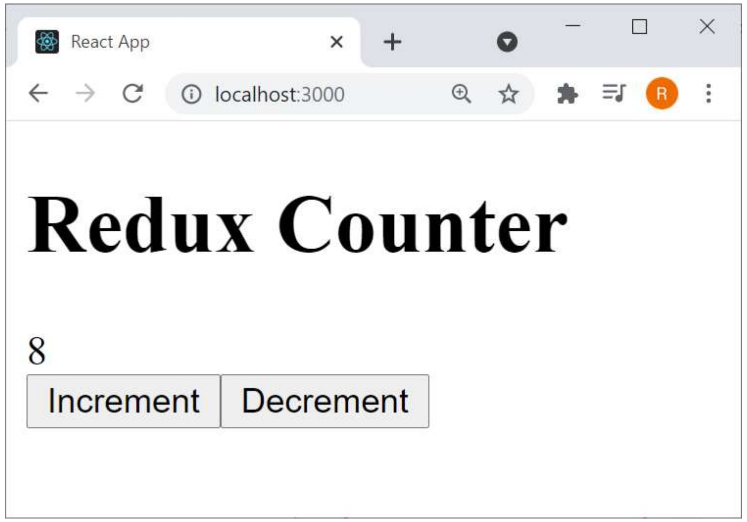
Why Redux?


Without Redux
With Redux
Store
Component initiating change

How does redux work?


Reducer function
Change data in the store
Action
Central State Store
Subscription to data in the store
Component
Dispatch an action to the reducer function

Simple counter



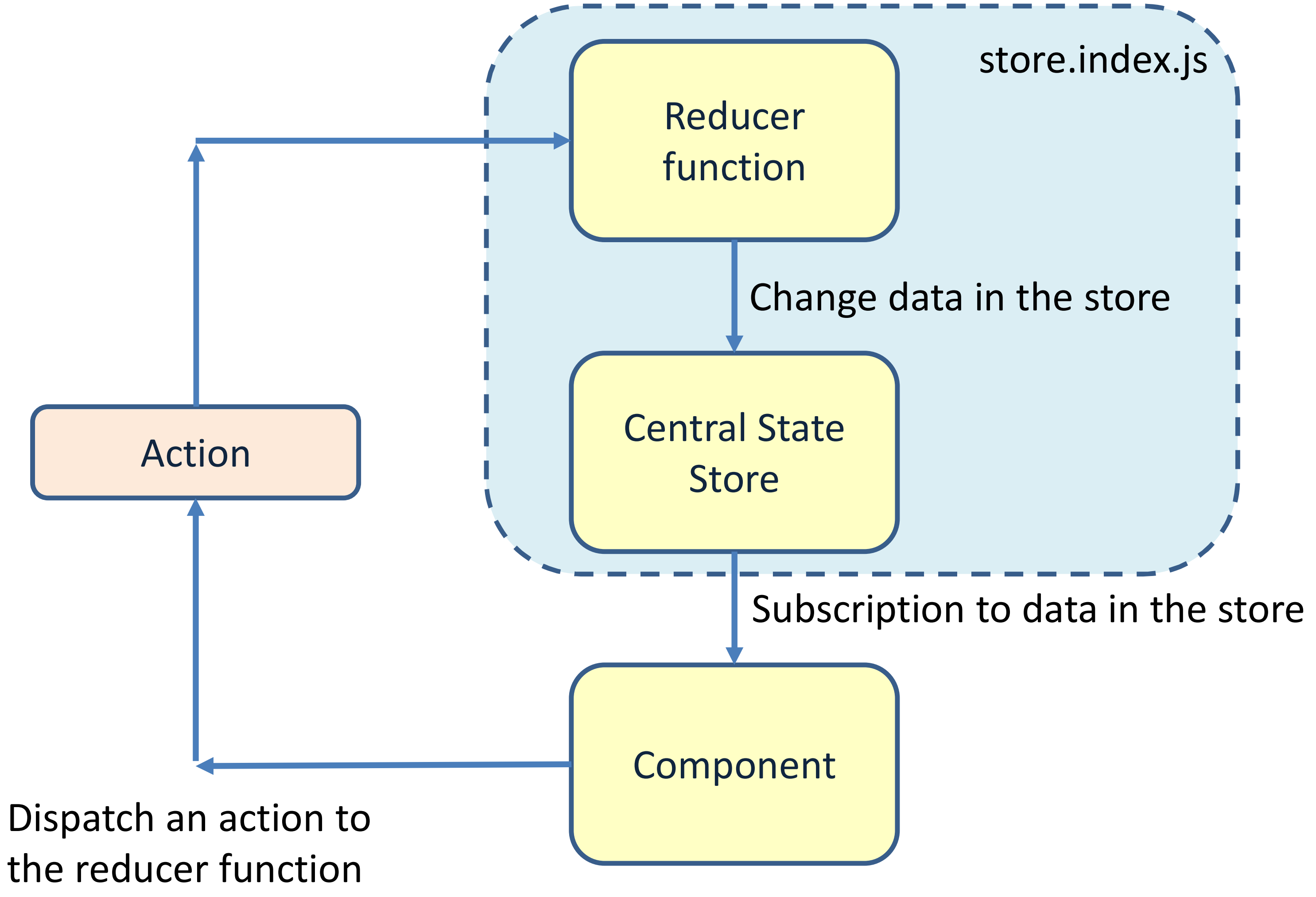
Redux Counter

8

Create a react redux app

1. Create a store
2. Make the store available to the whole app
3. Let the component use the store

1. Create the store



1. Create the store

Store/index.js

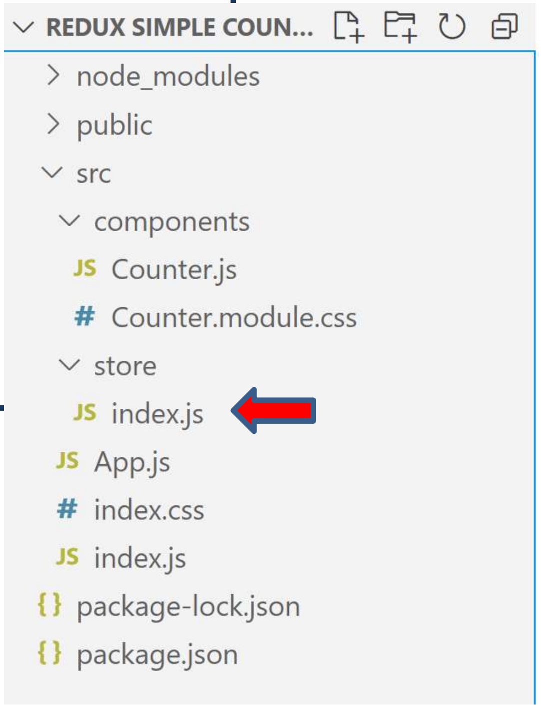
```
import {createStore} from 'redux';

const counterReducer = (state = {counter : 0 }, action ) => {
  if (action.type === 'increment'){
      return { counter : state.counter + 1};
  }
  if (action.type === 'decrement'){
    return { counter : state.counter - 1};
  }
  return state;
}

const store = createStore(counterReducer);

export default store;
```

Initial state

Reducer function for the store

Create the store

Export the store



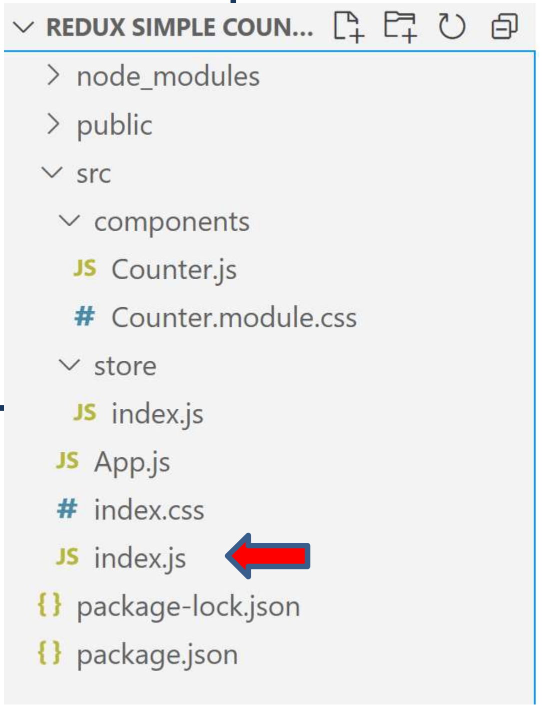
2. Make the store available to the whole app

index.js

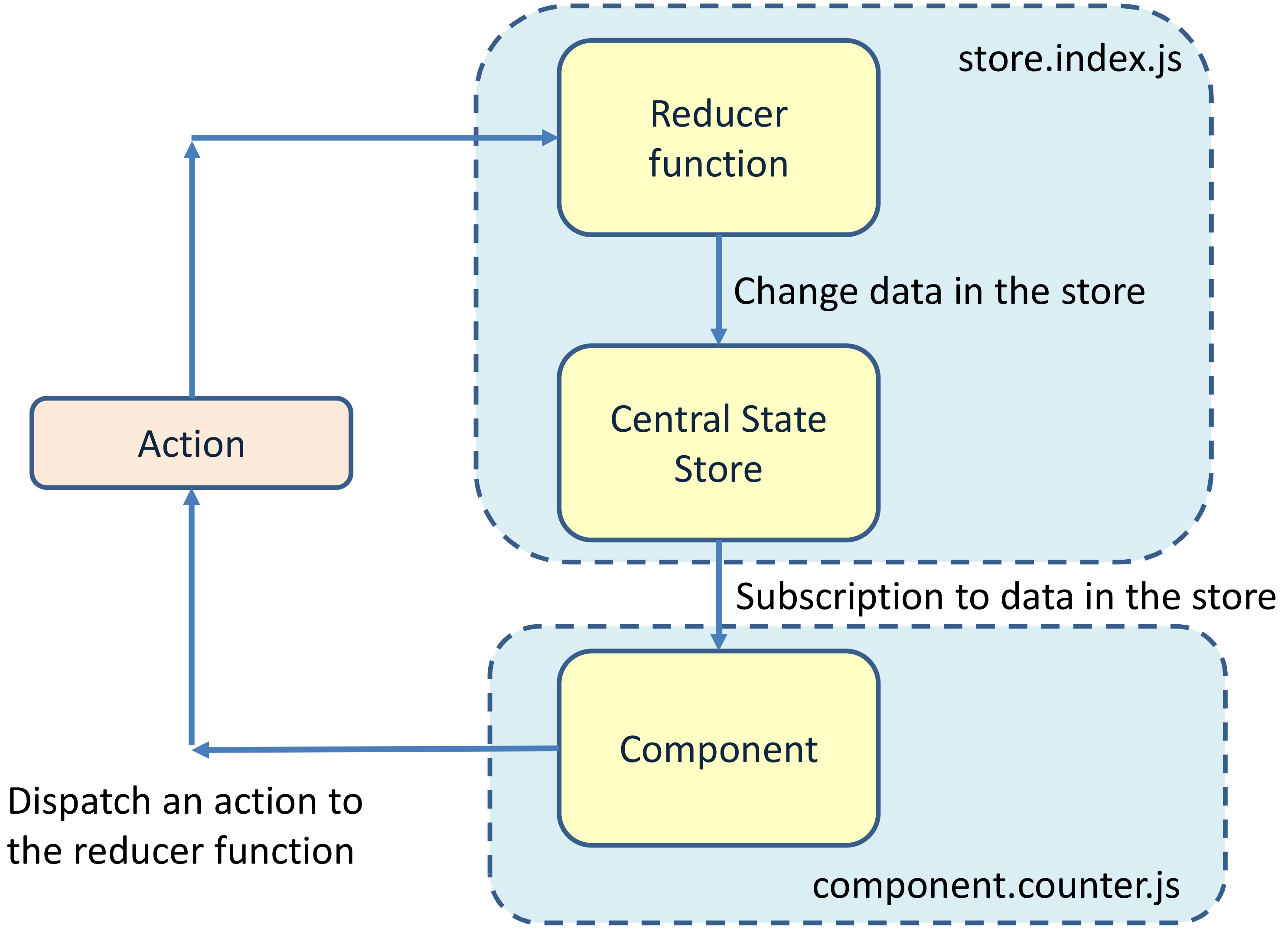
```
import React from 'react';
import ReactDOM from 'react-dom';
import { Provider } from 'react-redux'

import './index.css';
import App from './App';
import store from './store/index'

ReactDOM.render(
    <Provider store={store}>
        <App />
    </Provider>,
    document.getElementById('root')
);
```

Provide the store at the highest level of the application structure

Provide the store to all child components of the App



3. Let the component use the store



3. Let the component use the store

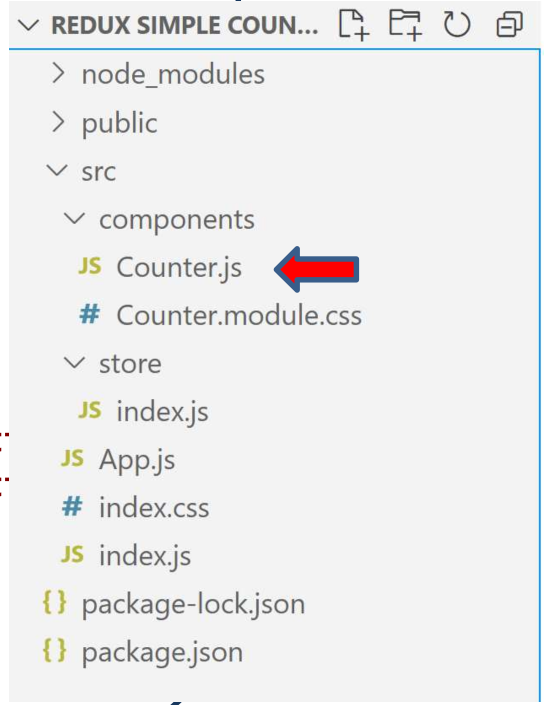
components/counter.js

```
import { useSelector, useDispatch } from 'react-redux';

const Counter = () => {
  const dispatch = useDispatch();
  const counter = useSelector(state => state.counter);

  const incrementHandler = () => {
    dispatch({ type : 'increment'});
  }
  const decrementHandler = () => {
    dispatch({ type : 'decrement'});
  }
  return (
    <div>
      <h1>Redux Counter</h1>
      <div>{counter}</div>
      <div>
        <button onClick={incrementHandler}>Increment</but
        <button onClick={decrementHandler}>Decrement</but
      </div>
    </div>
  );
};
export default Counter;
```

Select state.counter from the store

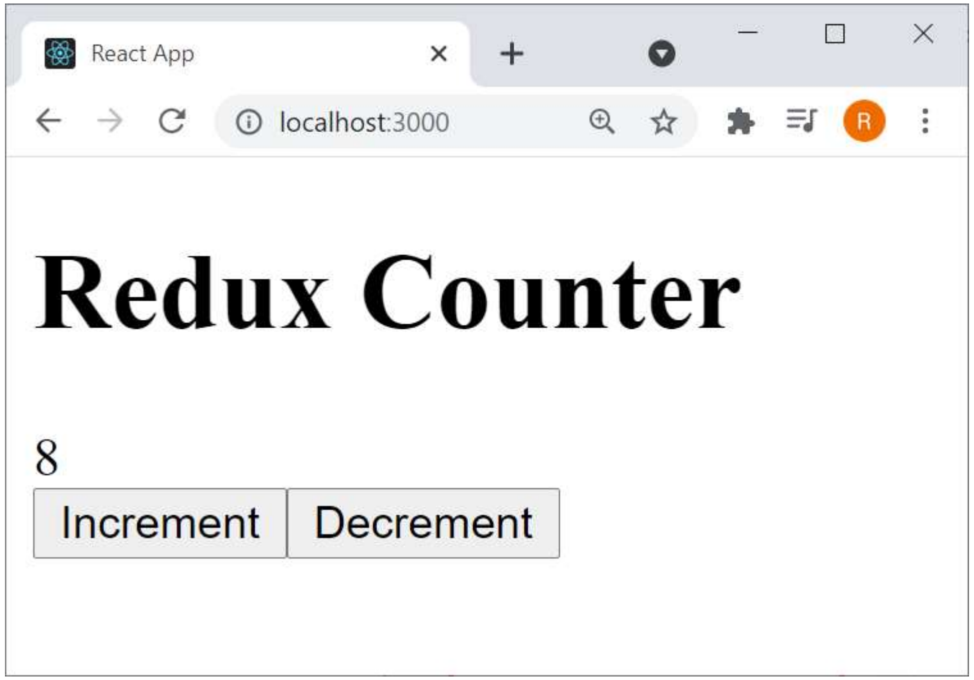
The counter value gets a subscription to the provided selector

Dispatch the 'decrement' action to the reducer function in the store



```
REDUX SIMPLE COUN...
  > node_modules
  > public
  v src
    v components
        Counter.js
        Counter.module.css
    v store
        index.js
      App.js
      index.css
      index.js
    package-lock.json
    package.json
```

Simple Counter app

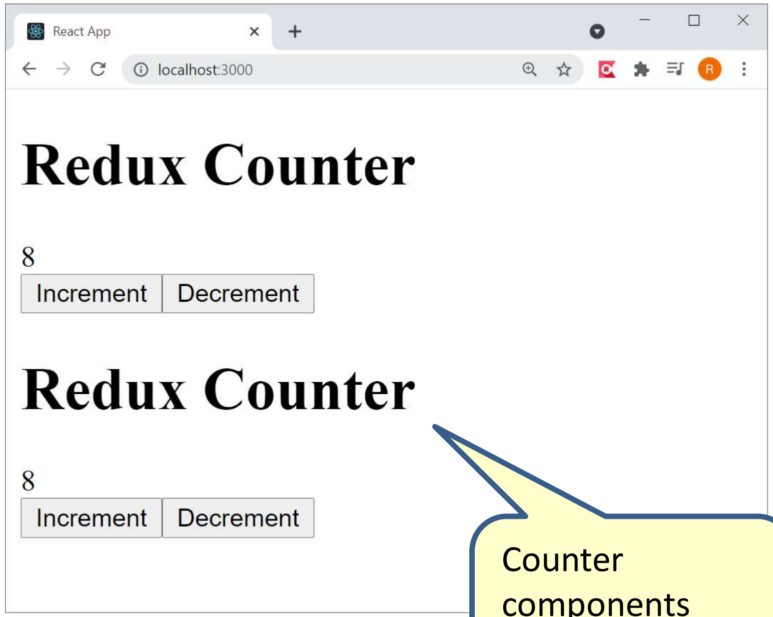


Redux Counter

8

2 Counter components

```
import Counter from './components/Counter';


function App() {
  return (
    <div>
      <Counter />
      <Counter />
    </div>
  );
}

export default App;
```



React App

localhost:3000

Redux Counter

8

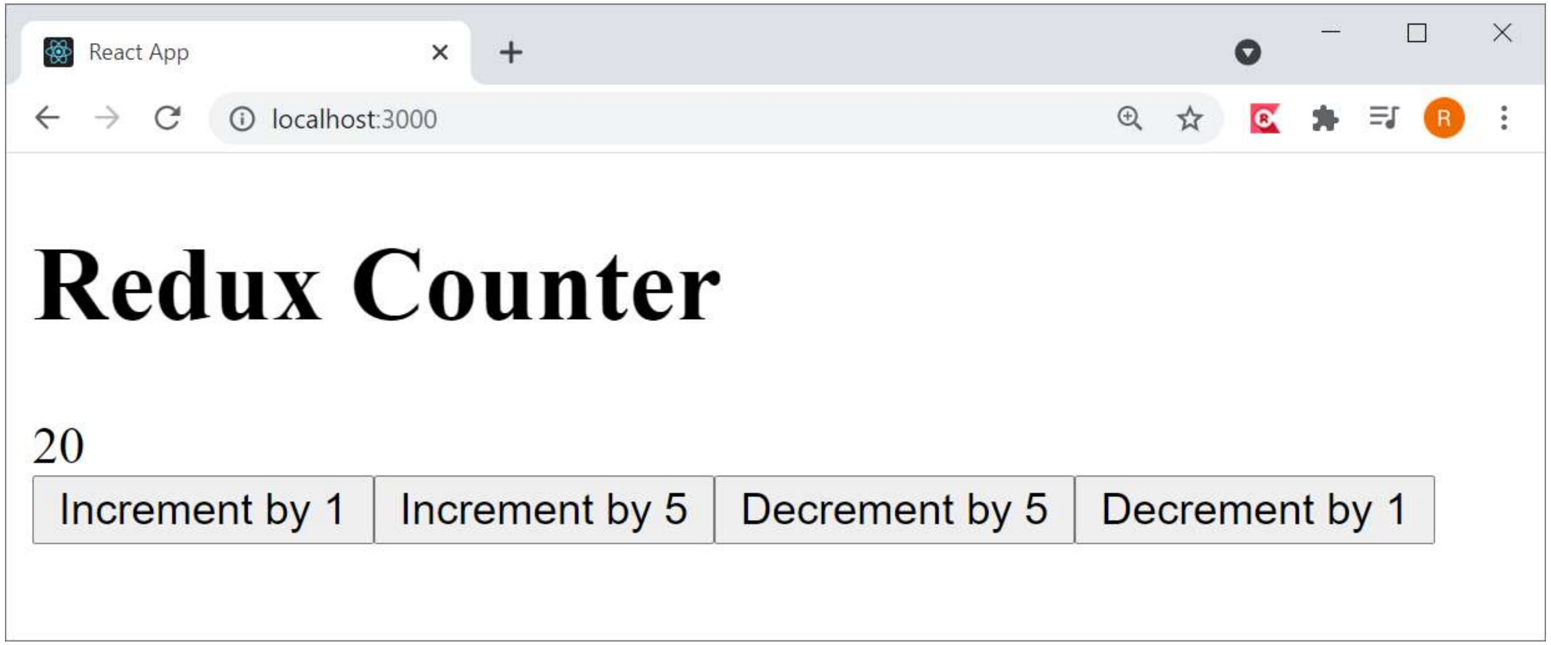
Redux Counter

8

Counter components share the same state

ADD A PAYLOAD TO THE ACTION

Counter with a value



Redux Counter

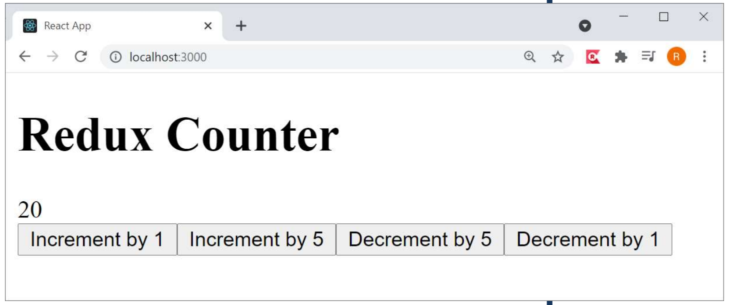
20

Counter.js

components/counter.js

```
const Counter = () => {
  const dispatch = useDispatch();
  const counter = useSelector(state => state.counter);
  const incrementHandler = () => {
    dispatch({ type : 'increment'});
  }
  const increaseHandler = () => {
    dispatch({ type : 'increase', amount : 5});
  }
  const decrementHandler = () => {
    dispatch({ type : 'decrement'});
  }
  const decreaseHandler = () => {
    dispatch({ type : 'decrease', amount : 5});
  }
  return (
    <div>
      <h1>Redux Counter</h1>
      <div>{counter}</div>
      <div>
        <button onClick={incrementHandler}>Increment by 1</button>
        <button onClick={increaseHandler}>Increment by 5</button>
        <button onClick={decreaseHandler}>Decrement by 5</button>
        <button onClick={decrementHandler}>Decrement by 1</button>
      </div>
    </div>
  );
```

Add a payload to the action



Redux Counter

20

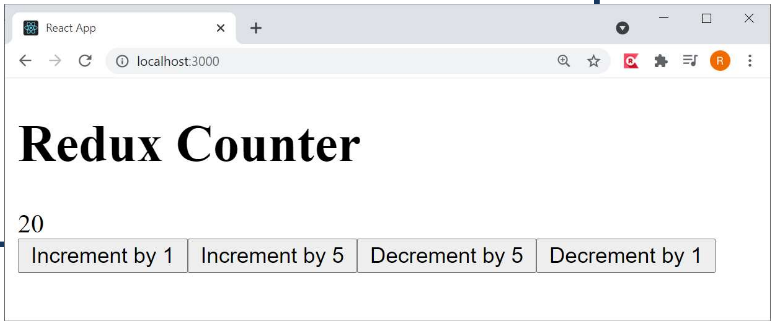
Increment by 1 | Increment by 5 | Decrement by 5 | Decrement by 1

index.js

components/counter.js

```
import { createStore } from 'redux';

const counterReducer = (state = { counter: 0 }, action) => {
  if (action.type === 'increment') {
    return { counter: state.counter + 1 };
  }
  if (action.type === 'increase') {
    return { counter: state.counter + action.amount };
  }
  if (action.type === 'decrement') {
    return { counter: state.counter - 1 };
  }
  if (action.type === 'decrease') {
    return { counter: state.counter - action.amount };
  }
  return state;
}

const store = createStore(counterReducer);

export default store;
```

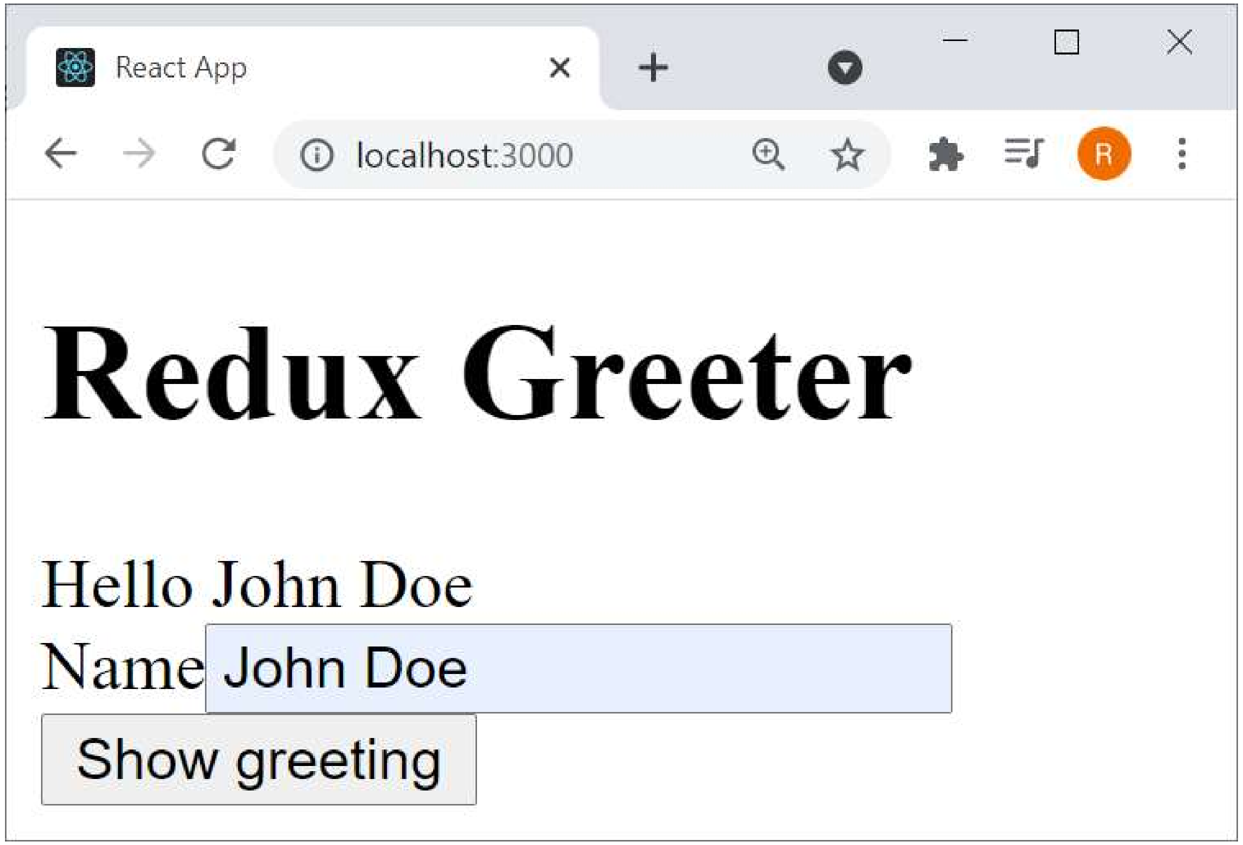
Get the payload from the action



Redux Counter

20

Simple Greeter App



Redux Greeter

Hello John Doe

Name John Doe

Greeter.js

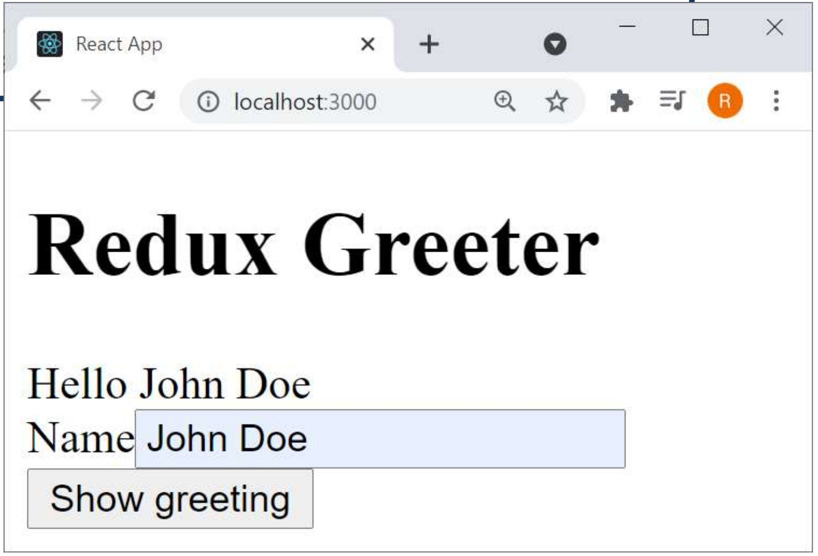
components/Greeter.js

```
import { useSelector, useDispatch } from 'react-redux';
import { useState } from 'react';
export const Greeter = () => {
  const [name, setName] = useState('');
  const dispatch = useDispatch();
  const greetingMessage = useSelector(state => state.greeting);
  const greetingHandler = () => {
    dispatch({ type : 'getgreeting', name : name });
  }
  return (
    <div>
      <h1>Redux Greeter</h1>
      <div>{greetingMessage}</div>
      <div>
      <div>
          Name
          <input
            type="text"
            placeholder="Your name"
            name="name"
            value={name}
            onChange={e => setName(e.target.value)} />
        </div>
        <button onClick={greetingHandler}>Show greeting</button>
      </div>
    </div>
  );
```

index.js

Initial state

```
import {createStore} from 'redux';

const reducer = (state = {greeting : 'Hello' }, action ) => {
  if (action.type === 'getgreeting'){
      return { greeting : "Hello "+action.name};
  }
  return state;
}

const store = createStore(reducer);

export default store;
```



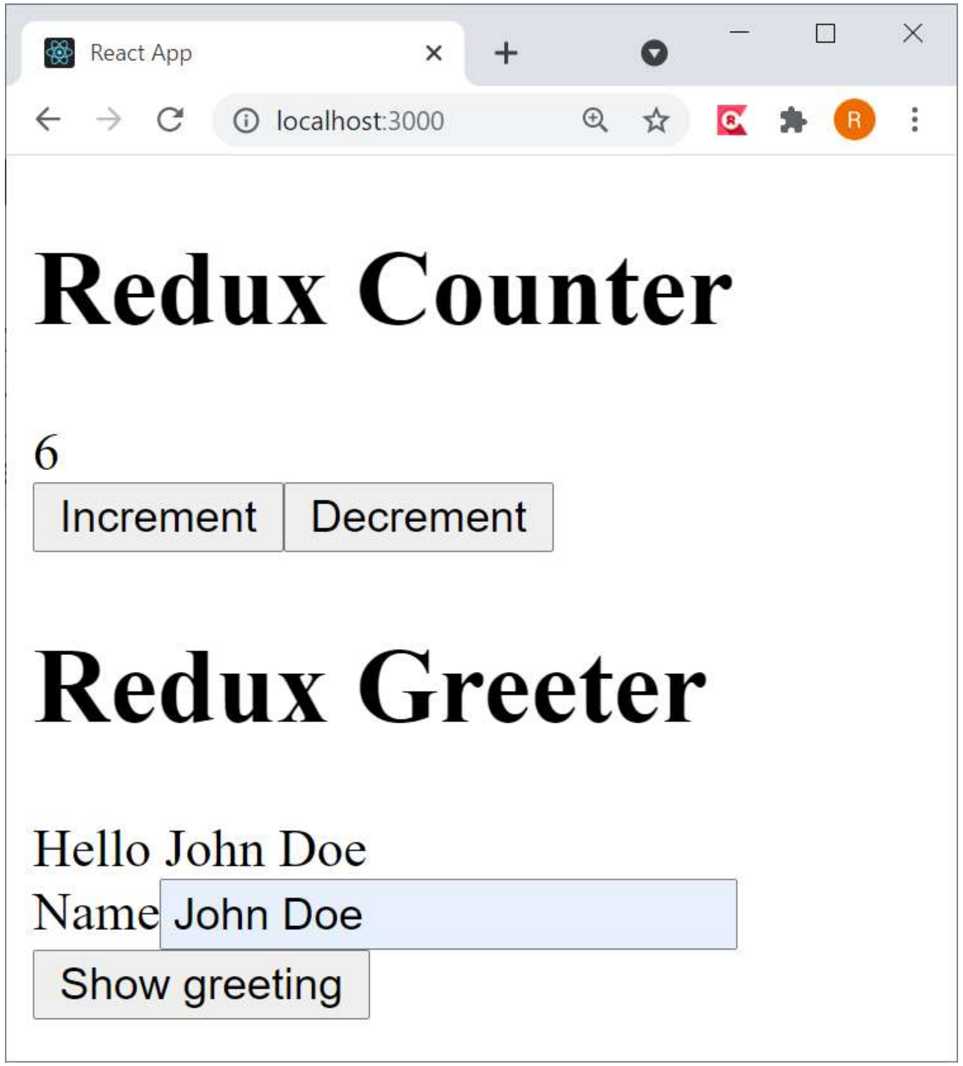
Redux Greeter

Hello John Doe

Name John Doe

MULTIPLE STATE PROPERTIES

Multiple state properties


React App

localhost:3000

Redux Counter

6

Redux Greeter

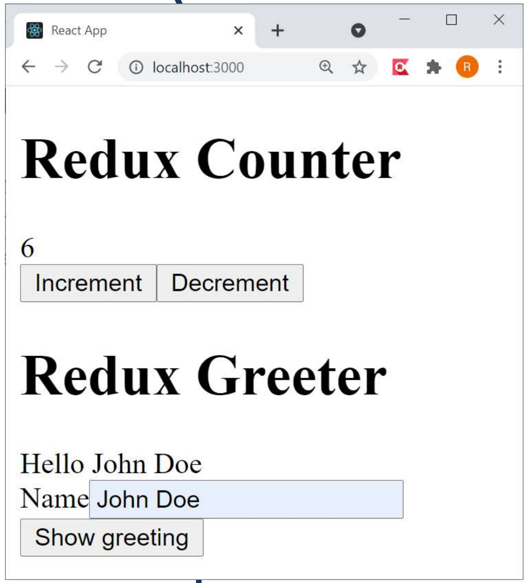
Hello John Doe

Name John Doe

Counter.js

components/Counter.js

```
import { useSelector, useDispatch } from 'react-redux';

const Counter = () => {
  const dispatch = useDispatch();
  const counter = useSelector(state => state.counter);

  const incrementHandler = () => {
    dispatch({ type : 'increment'});
  }
  const decrementHandler = () => {
    dispatch({ type : 'decrement'});
  }
  return (
    <div>
      <h1>Redux Counter</h1>
      <div>{counter}</div>
      <div>
        <button onClick={incrementHandler}>Increment</button>
        <button onClick={decrementHandler}>Decrement</button>
      </div>
    </div>
  );
};
export default Counter;
```



Redux Counter

6

Redux Greeter

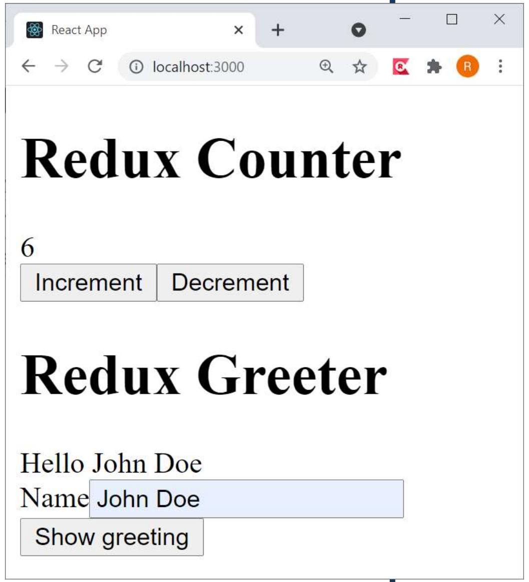
Hello John Doe

Name John Doe

Greeter.js

components/greeter.js

```
import { useSelector, useDispatch } from 'react-redux';
import { useState } from 'react';
export const Greeter = () => {
  const [name, setName] = useState('');
  const dispatch = useDispatch();
  const greetingMessage = useSelector(state => state.greeting);
  const greetingHandler = () => {
    dispatch({ type : 'getgreeting', name : name });
  }
  return (
    <div>
      <h1>Redux Greeter</h1>
      <div>{greetingMessage}</div>
      <div>
      <div>
          Name
          <input
            type="text"
            placeholder="Your name"
            name="name"
            value={name}
            onChange={e => setName(e.target.value)} />
        </div>
        <button onClick={greetingHandler}>Show greeting</button>
      </div>
    </div>
  );
```



Redux Counter

6

Redux Greeter

Hello John Doe

Name John Doe

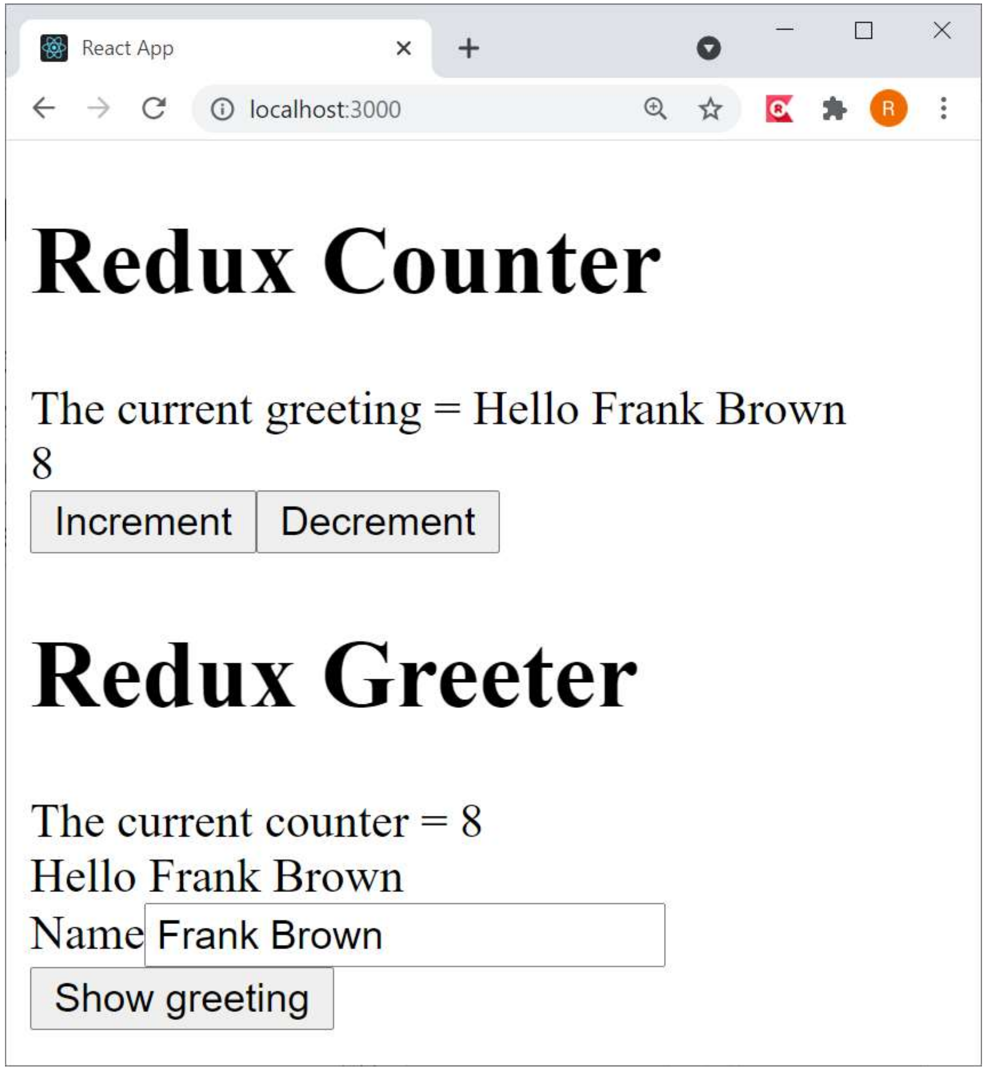
index.js

```
import { createStore } from 'redux';

const initalstate = { counter: 0, greeting: 'Hello' };
const counterReducer = (state = initalstate, action) => {
  if (action.type === 'increment') {
    return {
      counter: state.counter + 1,
      greeting: state.greeting
    };
  }
  if (action.type === 'decrement') {
    return {
      counter: state.counter - 1,
      greeting: state.greeting
    };
  }
  if (action.type === 'getgreeting') {
    return {
      counter: state.counter,
      greeting: "Hello " + action.name
    };
  }
  return state;
}
const store = createStore(counterReducer);
export default store;
```

State contains a counter and a greeting

ALWAYS return a newly created state

Get state from the store



Redux Counter

The current greeting = Hello Frank Brown
8

Redux Greeter

The current counter = 8
Hello Frank Brown
Name Frank Brown

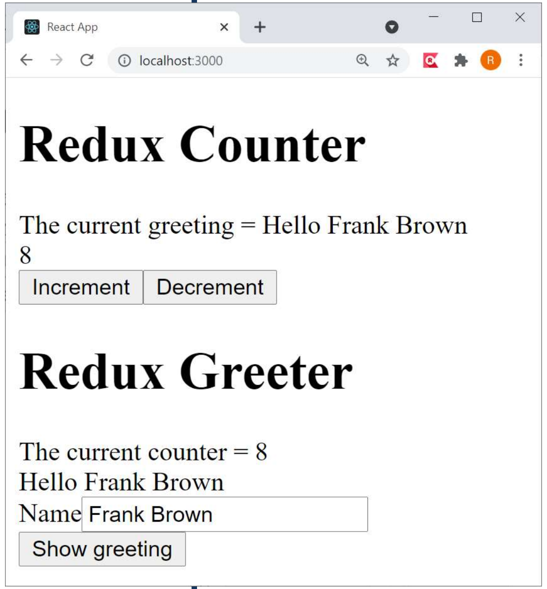
Counter.js

components/Counter.js

```
export const Counter = () => {
const dispatch = useDispatch();
const counter = useSelector(state => state.counter);
const greeting = useSelector(state => state.greeting);

const incrementHandler = () => {
  dispatch({ type : 'increment'});
}

const decrementHandler = () => {
  dispatch({ type : 'decrement'});
}

return (
  <div>
    <h1>Redux Counter</h1>
    <div>The current greeting = {greeting}</div>
    <div>{counter}</div>
    <div>
      <button onClick={incrementHandler}>Increment</button>
      <button onClick={decrementHandler}>Decrement</button>
    </div>
  </div>
  );
};
```

Get greeting state

Show greeting state



Redux Counter

The current greeting = Hello Frank Brown
8
Increment Decrement

Redux Greeter

The current counter = 8
Hello Frank Brown
Name Frank Brown
Show greeting

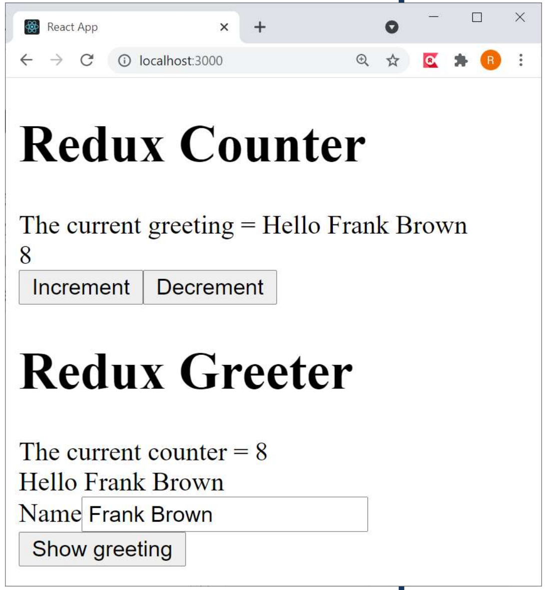
Greeter.js

components/Greeter.js

```
export const Greeter = () => {
const [name, setName] = useState('');
const dispatch = useDispatch();
const greetingMessage = useSelector(state => state.greeting);
const counter = useSelector(state => state.counter);

const greetingHandler = () => {
  dispatch({ type : 'getgreeting', name : name });
}
return (
  <div>
    <h1>Redux Greeter</h1>
    <div>The current counter = {counter}</div>
    <div>{greetingMessage}</div>
    <div>
    <div>
        Name
        <input
          type="text"
          placeholder="Your name"
          name="name"
          value={name}
          onChange={e => setName(e.target.value)} />
      </div>
      <button onClick={greetingHandler}>Show greeting</button>
    </div>
  </div>
```

Get counter state

Show counter state



React App

localhost:3000

Redux Counter

The current greeting = Hello Frank Brown
8

Redux Greeter

The current counter = 8
Hello Frank Brown
Name Frank Brown