

10. REDUX Examples - Counter.

Redux is distributed with a few examples in its source code. Most of these examples are also on **CodeSandbox**, an online editor that lets you play with the examples online.

10.1. Example 1 - Counter

It does not require a build system or a view framework and exists to show the raw Redux API used with ES5.

Example: A simple counter implemented using Redux.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Redux basic example</title>
    <script src="https://unpkg.com/redux@latest/dist/redux.min.js">
    </script>
  </head>
  <body>
    <div>
      <p>
        Clicked: <span id="value">0</span> times
        <button id="increment">+</button>
        <button id="decrement">-</button>
        <button id="incrementIfOdd">Increment if odd</button>
        <button id="incrementAsync">Increment async</button>
      </p>
    </div>
    <script>
      function counter(state, action) {
        if (typeof state === 'undefined') {
          return 0
        }

        switch (action.type) {
          case 'INCREMENT':
            return state + 1
          case 'DECREMENT':
            return state - 1
          default:
            return state
        }
      }

      var store = Redux.createStore(counter)
      var valueEl = document.getElementById('value')

      function render() {
        valueEl.innerHTML = store.getState().toString()
      }
    </script>
  </body>
</html>
```

```
    }

    render()
    store.subscribe(render)

    document.getElementById('increment')
      .addEventListener('click', function () {
        store.dispatch({ type: 'INCREMENT' })
      })

    document.getElementById('decrement')
      .addEventListener('click', function () {
        store.dispatch({ type: 'DECREMENT' })
      })

    document.getElementById('incrementIfOdd')
      .addEventListener('click', function () {
        if (store.getState() % 2 !== 0) {
          store.dispatch({ type: 'INCREMENT' })
        }
      })

    document.getElementById('incrementAsync')
      .addEventListener('click', function () {
        setTimeout(function () {
          store.dispatch({ type: 'INCREMENT' })
        }, 1000)
      })
  </script>
</body>
</html>
```

Output:

Redux Example - Counter Vanilla

Clicked: 5 times

<https://codesandbox.io/s/github/reduxjs/redux/tree/master/examples/counter-vanilla>

10.2. Example 2 – Counter with REACT + REDUX.

This is the most **basic example of using Redux together with React**. For simplicity, it re-renders the React component manually when the store changes. In real projects, you will likely want to use the highly performant React Redux bindings instead.

This example includes tests.

Example: index.js

```
import React from 'react'
import ReactDOM from 'react-dom'
import { createStore } from 'redux'
import Counter from '../components/Counter'
import counter from '../reducers'

const store = createStore(counter)
const rootEl = document.getElementById('root')

const render = () => ReactDOM.render(
  <Counter
    value={store.getState()}
    onIncrement={() => store.dispatch({ type: 'INCREMENT' })}
    onDecrement={() => store.dispatch({ type: 'DECREMENT' })}
  />,
  rootEl
)

render()
store.subscribe(render)
```

Example: reducers/index.js

```
export default (state = 0, action) => {
  switch (action.type) {
    case 'INCREMENT':
      return state + 1
    case 'DECREMENT':
      return state - 1
    default:
      return state
  }
}
```

Example: components/Counter.js

```
import React, { Component } from 'react'

class Counter extends Component {
  constructor(props) {
    super(props);
    this.incrementAsync = this.incrementAsync.bind(this);
    this.incrementIfOdd = this.incrementIfOdd.bind(this);
  }

  incrementIfOdd() {
    if (this.props.value % 2 !== 0) {
      this.props.onIncrement()
    }
  }

  incrementAsync() {
    setTimeout(this.props.onIncrement, 1000)
  }

  render() {
    const { value, onIncrement, onDecrement } = this.props
    return (
      <p>
        Clicked: {value} times
        {' '}
        <button onClick={onIncrement}>
          +
        </button>
        {' '}
        <button onClick={onDecrement}>
          -
        </button>
        {' '}
        <button onClick={this.incrementIfOdd}>
          Increment if odd
        </button>
        {' '}
        <button onClick={this.incrementAsync}>
          Increment async
        </button>
      </p>
    )
  }
}

export default Counter
```

Output:

React + Redux Basic Counter

Clicked: 3 times

<https://codesandbox.io/s/github/reduxjs/redux/tree/master/examples/counter>