**Example: Todo List**

This is the complete source code of the tiny todo app we built during the [basics tutorial](http://redux.js.org/docs/basics/index.html).

**Entry Point**

**index.js**

import React from 'react'

import { render } from 'react-dom'

import { Provider } from 'react-redux'

import { createStore } from 'redux'

import todoApp from './reducers'

import App from './components/App'

let store = createStore(todoApp)

render(

<Provider store={store}>

<App />

</Provider>,

document.getElementById('root')

)

**Action Creators**

**actions/index.js**

let nextTodoId = 0

export const addTodo = (text) => {

return {

type: 'ADD\_TODO',

id: nextTodoId++,

text

}

}

export const setVisibilityFilter = (filter) => {

return {

type: 'SET\_VISIBILITY\_FILTER',

filter

}

}

export const toggleTodo = (id) => {

return {

type: 'TOGGLE\_TODO',

id

}

}

**Reducers**

**reducers/todos.js**

const todo = (state, action) => {

switch (action.type) {

case 'ADD\_TODO':

return {

id: action.id,

text: action.text,

completed: false

}

case 'TOGGLE\_TODO':

if (state.id !== action.id) {

return state

}

return Object.assign({}, state, {

completed: !state.completed

})

default:

return state

}

}

const todos = (state = [], action) => {

switch (action.type) {

case 'ADD\_TODO':

return [

...state,

todo(undefined, action)

]

case 'TOGGLE\_TODO':

return state.map(t =>

todo(t, action)

)

default:

return state

}

}

export default todos

**reducers/visibilityFilter.js**

const visibilityFilter = (state = 'SHOW\_ALL', action) => {

switch (action.type) {

case 'SET\_VISIBILITY\_FILTER':

return action.filter

default:

return state

}

}

export default visibilityFilter

**reducers/index.js**

import { combineReducers } from 'redux'

import todos from './todos'

import visibilityFilter from './visibilityFilter'

const todoApp = combineReducers({

todos,

visibilityFilter

})

export default todoApp

**Presentational Components**

**components/Todo.js**

import React, { PropTypes } from 'react'

const Todo = ({ onClick, completed, text }) => (

<li

onClick={onClick}

style={{

textDecoration: completed ? 'line-through' : 'none'

}}

>

{text}

</li>

)

Todo.propTypes = {

onClick: PropTypes.func.isRequired,

completed: PropTypes.bool.isRequired,

text: PropTypes.string.isRequired

}

export default Todo

**components/TodoList.js**

import React, { PropTypes } from 'react'

import Todo from './Todo'

const TodoList = ({ todos, onTodoClick }) => (

<ul>

{todos.map(todo =>

<Todo

key={todo.id}

{...todo}

onClick={() => onTodoClick(todo.id)}

/>

)}

</ul>

)

TodoList.propTypes = {

todos: PropTypes.arrayOf(PropTypes.shape({

id: PropTypes.number.isRequired,

completed: PropTypes.bool.isRequired,

text: PropTypes.string.isRequired

}).isRequired).isRequired,

onTodoClick: PropTypes.func.isRequired

}

export default TodoList

**components/Link.js**

import React, { PropTypes } from 'react'

const Link = ({ active, children, onClick }) => {

if (active) {

return <span>{children}</span>

}

return (

<a href="#"

onClick={e => {

e.preventDefault()

onClick()

}}

>

{children}

</a>

)

}

Link.propTypes = {

active: PropTypes.bool.isRequired,

children: PropTypes.node.isRequired,

onClick: PropTypes.func.isRequired

}

export default Link

**components/Footer.js**

import React from 'react'

import FilterLink from '../containers/FilterLink'

const Footer = () => (

<p>

Show:

{" "}

<FilterLink filter="SHOW\_ALL">

All

</FilterLink>

{", "}

<FilterLink filter="SHOW\_ACTIVE">

Active

</FilterLink>

{", "}

<FilterLink filter="SHOW\_COMPLETED">

Completed

</FilterLink>

</p>

)

export default Footer

**components/App.js**

import React from 'react'

import Footer from './Footer'

import AddTodo from '../containers/AddTodo'

import VisibleTodoList from '../containers/VisibleTodoList'

const App = () => (

<div>

<AddTodo />

<VisibleTodoList />

<Footer />

</div>

)

export default App

**Container Components**

**containers/VisibleTodoList.js**

import { connect } from 'react-redux'

import { toggleTodo } from '../actions'

import TodoList from '../components/TodoList'

const getVisibleTodos = (todos, filter) => {

switch (filter) {

case 'SHOW\_ALL':

return todos

case 'SHOW\_COMPLETED':

return todos.filter(t => t.completed)

case 'SHOW\_ACTIVE':

return todos.filter(t => !t.completed)

}

}

const mapStateToProps = (state) => {

return {

todos: getVisibleTodos(state.todos, state.visibilityFilter)

}

}

const mapDispatchToProps = (dispatch) => {

return {

onTodoClick: (id) => {

dispatch(toggleTodo(id))

}

}

}

const VisibleTodoList = connect(

mapStateToProps,

mapDispatchToProps

)(TodoList)

export default VisibleTodoList

**containers/FilterLink.js**

import { connect } from 'react-redux'

import { setVisibilityFilter } from '../actions'

import Link from '../components/Link'

const mapStateToProps = (state, ownProps) => {

return {

active: ownProps.filter === state.visibilityFilter

}

}

const mapDispatchToProps = (dispatch, ownProps) => {

return {

onClick: () => {

dispatch(setVisibilityFilter(ownProps.filter))

}

}

}

const FilterLink = connect(

mapStateToProps,

mapDispatchToProps

)(Link)

export default FilterLink

**Other Components**

**containers/AddTodo.js**

import React from 'react'

import { connect } from 'react-redux'

import { addTodo } from '../actions'

let AddTodo = ({ dispatch }) => {

let input

return (

<div>

<form onSubmit={e => {

e.preventDefault()

if (!input.value.trim()) {

return

}

dispatch(addTodo(input.value))

input.value = ''

}}>

<input ref={node => {

input = node

}} />

<button type="submit">

Add Todo

</button>

</form>

</div>

)

}

AddTodo = connect()(AddTodo)

export default AddTodo