

## **react-redux-demo1: Working Steps**

**1) Copy the demo1-use-context-hook as react-redux-demo1**

**2) Install redux library**

```
npm install redux
```

**3) Install react-redux binding library**

```
npm install react-redux
```

**4) Define the Course Initial State (src\myredux\course\ courseReducer.js)**

```
const initialState = {  
  courseId: "C-11",  
  courseName: "Spring MicroServices",  
  cost: 18000,  
  trainer: "Srinivas Dande",  
}
```

**5) Define the Reducer to Return Initial State (courseReducer.js)**

```
const mycourseReducer = (state = initialState, action) => {  
  switch (action.type) {  
  
    default:  
      return state;  
  }  
}  
  
export default mycourseReducer;
```

**6) Write the Redux Store Configuration (src\myredux\course\ mystore.js)**

```
import { createStore, combineReducers } from 'redux';  
import mycourseReducer from './course/courseReducer';  
  
const rootReducer = combineReducers({  
  mycourse: mycourseReducer,  
});  
  
const mystore = createStore(rootReducer);  
  
export default mystore;
```

## 7) Update App.js with Redux Provider and Store

```
import { Provider } from 'react-redux';
import mystore from './myredux/mystore';

const App = () => {
  return (
    <Provider store={mystore}>
      <div className="text-center container">
        <h3> Welcome to React Redux </h3>
      </div>
    </Provider>
  );
}

export default App;
```

## 8) Run the Application

## 9) Write the Course Component

```
import { connect } from 'react-redux';

const CourseComponent = (props) => {

  const { courseId, courseName, cost, trainer } = props.mycourseData;

  return (
    <div className="container text-left">
      <h4> Course Id: {courseId}</h4>
      <h4> Course Name : {courseName}</h4>
      <h4> Cost : {cost}</h4>
      <h4> Trainer : {trainer}</h4>
    </div>
  );
}

const mapStateToProps = state => {
  return {
    mycourseData: state.mycourse
  }
}

export default connect(mapStateToProps)(CourseComponent);
```

10) **Include CourseComponent in App**

11) **Run the Application**

12) **Define the Student Initial State (src\myredux\student\studentReducer.js)**

```
const initialState = {  
  studentId: 101,  
  studentName: "Srinivas",  
  email: "Srinivas@jlc.com",  
  phone: 55555  
}
```

13) **13) Define the Reducer to Return Initial State (studentReducer.js)**

```
const mystudentReducer = (state = initialState, action) => {  
  switch (action.type) {  
  
    default:  
      return state;  
  }  
}  
  
export default mystudentReducer;
```

14) **Update Redux Store Configuration (src\myredux\mystore.js)**

```
import { createStore, combineReducers } from 'redux';  
  
import mycourseReducer from './course/courseReducer';  
import mystudentReducer from './student/studentReducer';  
  
const rootReducer = combineReducers({  
  mycourse: mycourseReducer,  
  mystudent: mystudentReducer  
});  
  
const mystore = createStore(rootReducer);  
  
export default mystore;
```

### 15) Write the Student Component

```
import { connect } from 'react-redux';
const StudentComponent = (props) => {

  const { studentId, studentName, email, phone } = props.mystudentData;
  return (
    <div className="container text-left ">
      <h4> Student ID : {studentId}</h4>
      <h4> Student Name : {studentName}</h4>
      <h4>Email Id: {email}</h4>
      <h4> Phone No : {phone}</h4>
    </div>
  );
}
const mapStateToProps = state => {
  return {
    mystudentData: state.mystudent
  }
}
export default connect(mapStateToProps)(StudentComponent);
```

### 16) Include StudentComponent in App

### 17) Run the Application

## react-redux-demo1: Working with React Redux

### react-redux-demo1: Files Required:

1. index.html	2. index.js
3. courseReducer.js	4. studentReducer.js
5. mystore.js	6. App.js
7. CourseComponent.js	8. StudentComponent.js
9. TestComponent.js	

#### 1. index.html

#### 2. index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';

ReactDOM.render(
  <App />,
  document.getElementById('myroot')
);
```



### 3. courseReducer.js

**//1.Define the Initial State**

```
const initialState = {  
  courseId: "C-104",  
  courseName: "ReEact JS 17",  
  cost: 13500,  
  trainer: "Srinivas Dande",  
}
```

**//2.Define the Reducer**

```
const mycourseReducer = (state = initialState, action) => {  
  switch (action.type) {  
  
    default:  
      return state;  
  }  
}
```

```
export default mycourseReducer;
```

### 4. studentReducer.js

**//1.Define the Initial State**

```
const initialState = {  
  studentId: 501,  
  studentName: "Dande",  
  email: "Dande@jlc.com",  
  phone: 99999  
}
```

**//2.Define the Reducer**

```
const mystudentReducer = (state = initialState, action) => {  
  switch (action.type) {  
  
    default:  
      return state;  
  }  
}
```

```
export default mystudentReducer;
```



## 5. mystore.js

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

import mycourseReducer from './course/courseReducer';
import mystudentReducer from './student/studentReducer';

const rootReducer = combineReducers({
  mycourse: mycourseReducer,
  mystudent: mystudentReducer
});

const mystore = createStore(rootReducer);

export default mystore;
```

## 6. App.js

```
import React, { Component } from "react";
import "bootstrap/dist/css/bootstrap.min.css";

import { Provider } from 'react-redux';
import mystore from './myredux/mystore';

import CourseComponent from './CourseComponent';
import StudentComponent from './StudentComponent';
class App extends Component {
  render() {
    return (
      <Provider store={mystore}>
        <div className="container">
          <h2> Welcome to JLC</h2>
          <hr/>
          <StudentComponent/>
          <hr/>
          <CourseComponent/>
        </div>
      </Provider>
    );
  }
}

export default App;
```

## 7. CourseComponent.js

```
import React from "react";

import { connect } from 'react-redux';

const CourseComponent = (props) => {

  const { courseId, courseName, cost, trainer } = props.mycourseData;

  return(
    <div className="container">
      <br />
      <b><u>
        <h2 className="text-left"> Course Info</h2> </u> </b>
      <br />
      <div className="text-left">
        <h3> Course Id: {courseId}</h3>
        <h3> Course Name : {courseName}</h3>
        <h3> Cost : {cost}</h3>
        <h3> Trainer : {trainer}</h3>
      </div>
    </div>

  );
}

const mapStateToProps = (state) => {
  return {
    mycourseData: state.mycourse
  }
}

export default connect(mapStateToProps)(CourseComponent);
```

## 8. StudentComponent.js

```
import React from "react";

import { connect } from 'react-redux';

const StudentComponent = (props) => {

  const { studentId, studentName, email, phone } = props.mystudentData;
```



```
return (  
  <div className="container">  
    <br />  
    <b><u> <h2 className="text-left"> Student Info</h2> </u> </b>  
    <br />  
    <div className="text-left">  
      <h4> Student ID : {studentId}</h4>  
      <h4> Student Name : {studentName}</h4>  
      <h4>Email Id: {email}</h4>  
      <h4> Phone No : {phone}</h4>  
    </div>  
  </div>  
  
  );  
}  
  
const mapStateToProps = state => {  
  return {  
    mystudentData: state.mystudent  
  }  
}  
  
export default connect(mapStateToProps)(StudentComponent);
```

## 9. TestComponent.js

```
import React from "react";  
  
import { connect } from 'react-redux';  
  
const TestComponent = (props) => {  
  
  const { studentId, studentName, email, phone } = props.mydata.mystudent;  
  const { courseId, courseName, cost, trainer } = props.mydata.mycourse;;  
  
  return (  
    <div className="container">  
      <br />  
      <b><u> <h2 className="text-left"> Student Info</h2> </u> </b> <br />  
      <div className="text-left">  
        <h4> Student ID : {studentId}</h4>  
        <h4> Student Name : {studentName}</h4>  
        <h4>Email Id: {email}</h4>  
        <h4> Phone No : {phone}</h4>  
      </div>  
    </div>  
  );  
}
```





```
<br />
<b><u>
  <h2 className="text-left"> Course Info</h2> </u> </b> <br />
<div className="text-left">
  <h3> Course Id: {courseId}</h3>
  <h3> Course Name : {courseName}</h3>
  <h3> Cost : {cost}</h3>
  <h3> Trainer : {trainer}</h3>
</div>
</div>

);
}

const mapStateToProps = state => {
  return {
    mydata: state
  }
}

export default connect(mapStateToProps)( TestComponent);
```

## **react-redux-demo2: Working with React Redux**

### **react-redux-demo2: Files Required:**

1. index.html	Same as demo1
2. index.js	Same as demo1
3. courseReducer.js	Same as demo1
4. studentReducer.js	Same as demo1
5. mystore.js	Same as demo1
6. App.js	Same as demo1
7. CourseComponent.js	<b>Updated in demo2</b>
8. StudentComponent.js	<b>Updated in demo2</b>
9. TestComponent.js	<b>Updated in demo2</b>

1. **index.html**
2. **index.js**
3. **courseReducer.js**
4. **studentReducer.js**
5. **mystore.js**
6. **App.js**

## 7. CourseComponent.js

```
import React from "react";
import { useSelector } from "react-redux";

const CourseComponent = (props) => {

  const mycourseData = useSelector(state => state.mycourse);
  const { courseId, courseName, cost, trainer } = mycourseData;
  return(
    <div className="container">
      <br /> <b><u>
        <h2 className="text-left"> Course Info</h2> </u> </b> <br />
      <div className="text-left">
        <h3> Course Id: {courseId}</h3>
        <h3> Course Name : {courseName}</h3>
        <h3> Cost : {cost}</h3>
        <h3> Trainer : {trainer}</h3>
      </div>
    </div>
  );
}

export default CourseComponent;
```

## 8. StudentComponent.js

```
import React from "react";
import { useSelector } from "react-redux";

const StudentComponent = (props) => {

  const mystudentData = useSelector(state => state.mystudent);
  const { studentId, studentName, email, phone } = mystudentData;
  return (
    <div className="container">
      <br /> <b><u> <h2 className="text-left"> Student Info</h2> </u> </b> <br />
      <div className="text-left">
        <h4> Student ID : {studentId}</h4>
        <h4> Student Name : {studentName}</h4>
        <h4>Email Id: {email}</h4>
        <h4> Phone No : {phone}</h4>
      </div>
    </div>
  );
}

export default StudentComponent;
```



## 9. TestComponent.js

```
import React from "react";

import { useSelector } from "react-redux";

const TestComponent = (props) => {

  const mydata = useSelector(state => state);

  const { studentId, studentName, email, phone } = mydata.mystudent;
  const { courseId, courseName, cost, trainer } = mydata.mycourse;

  return (
    <div className="container">
      <br />
      <b><u> <h2 className="text-left"> Student Info</h2> </u> </b>
      <br />
      <div className="text-left">
        <h4> Student ID : {studentId}</h4>
        <h4> Student Name : {studentName}</h4>
        <h4>Email Id: {email}</h4>
        <h4> Phone No : {phone}</h4>
      </div>

      <br />
      <b><u>
        <h2 className="text-left"> Course Info</h2> </u> </b>
      <br />
      <div className="text-left">
        <h3> Course Id: {courseId}</h3>
        <h3> Course Name : {courseName}</h3>
        <h3> Cost : {cost}</h3>
        <h3> Trainer : {trainer}</h3>
      </div>
    </div>

  );
}
```

export default TestComponent;

## **react-redux-demo3: Working with React Redux**

### **react-redux-demo3: Files Required:**

1. index.html	2. index.js
3. courseTypes.js	4. courseActions.js
5. courseReducer.js	6. mystore.js
7. App.js	8. CourseComponent.js

#### **1. index.html**

#### **2. index.js**

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';

ReactDOM.render(
  <App />,
  document.getElementById('myroot')
);
```

#### **3. courseTypes.js**

```
export const UPDATE_CNAME="UPDATE_CNAME";
export const UPDATE_PRICE="UPDATE_PRICE";
export const UPDATE_DURATION="UPDATE_DURATION";
export const UPDATE_TRAINER="UPDATE_TRAINER";
```

#### **4. courseActions.js**

```
//Define Action Creators
import {UPDATE_CNAME} from './courseTypes';
import {UPDATE_PRICE} from './courseTypes';
import {UPDATE_DURATION} from './courseTypes';
import {UPDATE_TRAINER} from './courseTypes';

export const updateCname = (updatedCname)=> {
  return {
    type:UPDATE_CNAME,
    payload:updatedCname,
  };
}

export const updatePrice = (updatedPrice)=> {
  return {
    type:UPDATE_PRICE,
    payload:updatedPrice
  };
}
```



```
export const updateDuration = (updatedDuration)=> {  
  return {  
    type:UPDATE_DURATION,  
    payload:updatedDuration  
  };  
}
```

```
export const updateTrainer = (updatedTrainer)=> {  
  return {  
    type:UPDATE_TRAINER,  
    payload:updatedTrainer  
  };  
}
```

## 5. courseReducer.js

```
import {UPDATE_CNAME} from './courseTypes';  
import {UPDATE_PRICE} from './courseTypes';  
import {UPDATE_DURATION} from './courseTypes';  
import {UPDATE_TRAINER} from './courseTypes';
```

### //1.Define the Initial State

```
const initialState= {  
  courseId: "C-101",  
  courseName: "Master React ",  
  price: 10000,  
  duration:"50 Hrs",  
  trainer: "SD"  
}
```

### //2.Define the Reducer

```
const mycourseReducer = (currentState=initialState,action) => {  
  switch(action.type){  
  
    case UPDATE_CNAME:  
      return {  
        ...currentState,  
        courseName:action.payload  
      };  
  
    case UPDATE_PRICE:  
      return {  
        ...currentState,  
        price:action.payload  
      };  
  }
```

**case UPDATE\_DURATION:**

```
return {
  ...currentState,
  duration:action.payload
};
```

**case UPDATE\_TRAINER:**

```
return {
  ...currentState,
  trainer:action.payload
};
```

**default:**

```
return currentState
}
}
```

export default mycourseReducer;

## 6. **mystore.js**

```
import { createStore, combineReducers } from 'redux';
import mycourseReducer from './course/courseReducer';
```

```
const rootReducer = combineReducers({
  mycourse: mycourseReducer,
});
```

```
const mystore = createStore(rootReducer);
```

export default mystore;

## 7. **App.js**

```
import React, { Component } from "react";
```

```
import "bootstrap/dist/css/bootstrap.min.css";
```

```
import { Provider } from 'react-redux';
import mystore from './myredux/mystore';
```

```
import CourseComponent from './CourseComponent';
```

```
class App extends Component {
  render() {
    return (
```



```
<Provider store={mystore}>
  <div className="container">
    <h2> Welcome to JLC</h2>
    <hr/>
    <CourseComponent/>
  </div>
</Provider>
);
}
```

```
export default App;
```

## 8. CourseComponent.js

```
import React from "react";
```

```
import { connect } from "react-redux";
```

```
import { updateCname } from './myredux/course/courseActions';
```

```
import { updatePrice } from './myredux/course/courseActions';
```

```
import { updateDuration } from './myredux/course/courseActions';
```

```
import { updateTrainer } from './myredux/course/courseActions';
```

```
const CourseComponent = (props) => {
```

```
  const { courseId, courseName, price, duration, trainer } = props.mycourseData;
```

```
  return(
```

```
    <div className="container">
```

```
      <br /> <b><u>
```

```
      <h3 className="text-center"> Course Info</h3> </u> </b> <br />
```

```
      <div>
```

```
        <table className="table">
```

```
          <tbody>
```

```
            <tr>
```

```
              <td><h4> Course Id: {courseId}</h4> </td>
```

```
              <td> </td>
```

```
            </tr>
```

```
            <tr>
```

```
              <td><h4> Course Name : {courseName}</h4> </td>
```

```
              <td>
```

```
                <button onClick={() => props.courseDispatch(updateCname("Master React 17.1"))}>
```

```
                className="btn btn-success">
```

```
                Update Course Name
```

```
                </button>
```

```
              </td>
```

```
            </tr>
```



```
<tr> <td><h4> Price : {price}</h4> </td>
<td>
<button onClick={() => props.courseDispatch(updatePrice(13200))}
className="btn btn-success">
Update Price
</button>
</td>
</tr>
<tr> <td><h4> Duration : {duration}</h4> </td>
<td>
<button onClick={() => props.courseDispatch(updateDuration("79 Hrs"))}
className="btn btn-success">
Update Duration
</button>
</td>
</tr>
<tr> <td><h4> Trainer : {trainer}</h4> </td>
<td>
<button onClick={() => props.courseDispatch(updateTrainer("Srinivas Dande"))}
className="btn btn-success">
Update Trainer
</button>
</td>
</tr>
</tbody>
</table>
</div>
</div>
);
}

const mapState = (state) => {
return {
mycourseData: state.mycourse
}
}

const mapDispatch = (courseDispatch) => {
return {
courseDispatch:courseDispatch
}
}

export default connect(mapState, mapDispatch)(CourseComponent);
```



## **react-redux-demo4: Working with React Redux**

### **react-redux-demo4: Files Required:**

1. index.html	Same as demo3
2. index.js	Same as demo3
3. courseTypes.js	Same as demo3
4. courseActions.js	Same as demo3
5. courseReducer.js	Same as demo3
6. mystore.js	Same as demo3
7. App.js	Same as demo3
8. CourseComponent.js	<b>Updated in demo4</b>

#### **1. index.html**

#### **2. index.js**

#### **3. courseReducer.js**

#### **4. studentReducer.js**

#### **5. mystore.js**

#### **6. App.js**

#### **7. CourseComponent.js**

```
import React from "react";
```

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

```
import { updateCname } from './myredux/course/courseActions';
```

```
import { updatePrice } from './myredux/course/courseActions';
```

```
import { updateDuration } from './myredux/course/courseActions';
```

```
import { updateTrainer } from './myredux/course/courseActions';
```

```
const CourseComponent = () => {
```

```
  const mycourseData = useSelector(state => state.mycourse);
```

```
  const { courseId, courseName, price, duration, trainer } = mycourseData;
```

```
  const courseDispatch = useDispatch();
```

```
  return(
```

```
    <div className="container">
```

```
      <br /> <b><u>
```

```
      <h3 className="text-center"> Course Info</h3> </u> </b>    <br />
```

```
      <div>
```

```
        <table className="table">
```

```
          <tbody>
```

```
            <tr>
```

```
              <td><h4> Course Id: {courseId}</h4> </td>
```



```
<td> </td>
</tr>
<tr>
<td><h4> Course Name : {courseName}</h4> </td>
<td>
<button onClick={() => courseDispatch(updateCname("Learn React 17"))}
className="btn btn-success">
Update Course Name
</button>
</td>
</tr>
<tr> <td><h4> Price : {price}</h4> </td>
<td>
<button onClick={() => courseDispatch(updatePrice(13500))} className="btn btn-
success">
Update Price
</button>
</td>
</tr>
<tr> <td><h4> Duration : {duration}</h4> </td>
<td>
<button onClick={() => courseDispatch(updateDuration("80 Hrs"))}
className="btn btn-success">
Update Duration
</button>
</td>
</tr>
<tr> <td><h4> Trainer : {trainer}</h4> </td>
<td>
<button onClick={() => courseDispatch(updateTrainer("Srinivas Dande"))}
className="btn btn-success">
Update Trainer
</button>
</td>
</tr>
</tbody>
</table>
</div>
</div>
);
}
```

```
export default CourseComponent;
```

## **react-redux-demo5: Working Steps**

**1) Copy the react-redux-demo4 as react-redux-demo5**

**2) Install redux-logger**

```
npm install redux-logger
```

**3) Install redux-thunk**

```
npm install redux-thunk
```

**4) Define the Course Initial State (src\myredux\course\ courseReducer.js)**

```
const allCoursesInitialState = {  
  loadingStatus: false,  
  courseList: [],  
  errorMessgae: {}  
}
```

**5) Define the Action Types (src\myredux\course\courseTypes.js)**

```
export const FETCH_COURSES_REQUEST = "FETCH_COURSES_REQUEST";  
export const FETCH_COURSES_SUCCESS = "FETCH_COURSES_SUCCESS";  
export const FETCH_COURSES_FAILURE = "FETCH_COURSES_FAILURE";
```

**6) Define the Reducer with Actions (src\myredux\course\courseReducer.js)**

**7) Define the Action Creators (src\myredux\course\courseActions.js)**

**8) Write the Redux Store Configuration (src\myredux\mystore.js)**

**9) Update App Component with Redux Provider and Store**

**10) Run the Application**

**11) Write the CourseListComponent**

**12) Include CourseListComponent in App**

**13) Run the Application**

## **react-redux-demo5: Working with React Redux**

### **react-redux-demo5: Files Required:**

1. index.html	2. index.js
3. courseTypes.js	4. courseActions.js
5. courseReducer.js	6. mystore.js
7. App.js	8. CourseListComponent.js

#### **1. index.html**

#### **2. index.js**

```
import React from "react";
import ReactDOM from "react-dom";
import App from "./App";
```

```
ReactDOM.render(<App />, document.getElementById("myroot"));
```

#### **3. courseTypes.js**

```
export const FETCH_COURSES_REQUEST = "FETCH_COURSES_REQUEST";
export const FETCH_COURSES_SUCCESS = "FETCH_COURSES_SUCCESS";
export const FETCH_COURSES_FAILURE = "FETCH_COURSES_FAILURE";
```

#### **4. courseActions.js**

```
//Define Action Creators
import { FETCH_COURSES_REQUEST } from './courseTypes';
import { FETCH_COURSES_SUCCESS } from './courseTypes';
import { FETCH_COURSES_FAILURE } from './courseTypes';

import axios from 'axios';

export const fetchCourseRequest = () => {
  return {
    type: FETCH_COURSES_REQUEST,
  }
}

export const fetchCourseSuccess = (courses) => {
  return {
    type: FETCH_COURSES_SUCCESS,
    payload: courses
  }
}

export const fetchCourseFailure = (error) => {
  return {
    type: FETCH_COURSES_FAILURE,
    payload: error
  }
}
```

```
export const fetchAllCourses = () => {

  return (courseDispatch) => {

    courseDispatch(fetchCourseRequest()) //1

    axios.get("https://coursecube.com/jlc-rest-api/mini-courses")
      .then(response => {
        const courseList = response.data;
        courseDispatch(fetchCourseSuccess(courseList))
      })
      .catch(error => {
        courseDispatch(fetchCourseFailure(error))
      })
  }
}
```

## 5. courseReducer.js

```
import { FETCH_COURSES_REQUEST } from './courseTypes';
import { FETCH_COURSES_SUCCESS } from './courseTypes';
import { FETCH_COURSES_FAILURE } from './courseTypes';
```

### //1. Define the Initial State

```
const allCoursesInitialState = {
  loadingStatus: false,
  courseList: [],
  error: {}
}
```

### //2. Define the Reducer

```
const myAllCoursesReducer = (currentState=allCoursesInitialState,action) => {
  switch(action.type){

    case FETCH_COURSES_REQUEST:
      return {
        loadingStatus: true,
        courseList: [],
        error: {}
      }
    case FETCH_COURSES_SUCCESS:
      return {
        loadingStatus: false,
        courseList: action.payload,
        error: {}
      }
  }
}
```



```
case FETCH_COURSES_FAILURE:
  return {
    loadingStatus: false,
    courseList: [],
    error: action.payload
  }
default:
  return currentState
}
}
```

export default myAllCoursesReducer;

## 6. mystore.js

```
import { createStore, combineReducers, applyMiddleware } from 'redux';

import logger from 'redux-logger';
import thunk from 'redux-thunk';

import myAllCoursesReducer from './course/courseReducer';

const rootReducer = combineReducers({
  mycourses: myAllCoursesReducer,
});

const mystore = createStore(rootReducer, applyMiddleware(logger, thunk));

export default mystore;
```

## 7. App.js

```
import React, { Component } from "react";
import "bootstrap/dist/css/bootstrap.min.css";

import { Provider } from 'react-redux';
import mystore from './myredux/mystore';

import CourseListComponent from './CourseListComponent';

class App extends Component {
  render() {
    return (
      <Provider store={mystore}>
        <div className="container">
          <h2> Welcome to JLC</h2>
          <hr/>

```



```
    <CourseListComponent/>
  </div>
</Provider>
);
}
}
```

```
export default App;
```

## 8. CourseListComponent.js

```
import React, { useEffect } from "react";

import { connect } from 'react-redux';
import { fetchAllCourses } from './myredux/course/courseActions';

const CourseListComponent = ({ mycourseData, courseDispatch }) => {

  useEffect(() => {
    courseDispatch(fetchAllCourses());
  }, []);

  let messageToDisplay = null;
  if (mycourseData.loadingStatus) {
    messageToDisplay = "Loading .....Please Wait.... ";
  } else if (mycourseData.error) {
    messageToDisplay = mycourseData.error.message;
  }

  let coursesToDislay = null;
  if (mycourseData && mycourseData.courseList) {
    coursesToDislay = mycourseData.courseList;
  }

  const mycourseList = coursesToDislay.map(
    mycourse => {
      return (
        <tr key={mycourse.courseId}>
          <td> {mycourse.courseId} </td>
          <td> {mycourse.courseName} </td>
          <td> {mycourse.duration} </td>
          <td> {mycourse.trainer} </td>
          <td> {mycourse.enrollments} </td>
        </tr>
      );
    }
  );
};
```



```
return (  
  <div className="container">  
  
    <table className="table">  
      <thead>  
        <tr>  
          <th> Course ID</th>  
          <th> Course Name</th>  
          <th> Duration</th>  
          <th> Trainer</th>  
          <th> Enrollments</th>  
          <th> </th>  
        </tr>  
      </thead>  
      <tbody>  
        {mycourseList}  
      </tbody>  
    </table>  
    <h2 className="text-center"> {messageToDisplay} </h2>  
  
  </div>  
</div>  
);  
}  
  
const mapState = state => {  
  return {  
    mycourseData: state.mycourses  
  }  
}  
  
const mapDispatch = dispatch => {  
  return {  
    courseDispatch : dispatch,  
  }  
}  
  
export default connect(mapState, mapDispatch)(CourseListComponent);
```



## react-redux-demo6: Working with React Redux

### react-redux-demo6: Files Required:

1. index.html	2. index.js
3. courseTypes.js	4. courseActions.js
5. courseReducer.js	6. mystore.js
7. App.js	8. JLCHeader.js
9. JLCBody.js	10. CourseListComponent.js
11. CourseDetailsComponent.js	

#### 1. index.html

#### 2. index.js

```
import React from "react";
import ReactDOM from "react-dom";
import App from "./App";
```

```
ReactDOM.render(<App />, document.getElementById("myroot"));
```

#### 3. courseTypes.js

```
export const FETCH_COURSES_REQUEST = "FETCH_COURSES_REQUEST";
export const FETCH_COURSES_SUCCESS = "FETCH_COURSES_SUCCESS";
export const FETCH_COURSES_FAILURE = "FETCH_COURSES_FAILURE";
```

```
export const FETCH_ONE_COURSE_REQUEST = "FETCH_ONE_COURSE_REQUEST";
export const FETCH_ONE_COURSE_SUCCESS = "FETCH_ONE_COURSE_SUCCESS";
export const FETCH_ONE_COURSE_FAILURE = "FETCH_ONE_COURSE_FAILURE";
```

#### 4. courseActions.js

```
import { FETCH_COURSES_REQUEST } from './courseTypes';
import { FETCH_COURSES_SUCCESS } from './courseTypes';
import { FETCH_COURSES_FAILURE } from './courseTypes';
```

```
import { FETCH_ONE_COURSE_REQUEST } from './courseTypes';
import { FETCH_ONE_COURSE_SUCCESS } from './courseTypes';
import { FETCH_ONE_COURSE_FAILURE } from './courseTypes';
```

```
import axios from 'axios';
```

```
export const fetchCourseRequest = () => {
  return {
    type: FETCH_COURSES_REQUEST,
  }
}
```



```
export const fetchCourseSuccess = (courses) => {
  return {
    type: FETCH_COURSES_SUCCESS,
    payload: courses
  }
}
export const fetchCourseFailure = (error) => {
  return {
    type: FETCH_COURSES_FAILURE,
    payload: error
  }
}
export const fetchCourses = () => {
  return (dispatch) => {

    dispatch(fetchCourseRequest())

    axios.get("https://coursecube.com/jlc-rest-api/mini-courses")
      .then(response => {
        const courseList = response.data;
        dispatch(fetchCourseSuccess(courseList))
      })
      .catch(error => {
        dispatch(fetchCourseFailure(error))
      })

  }
}

export const fetchOneCourseRequest = () => {
  return {
    type: FETCH_ONE_COURSE_REQUEST,
  }
}
export const fetchOneCourseSuccess = (course) => {
  return {
    type: FETCH_ONE_COURSE_SUCCESS,
    payload: course
  }
}
export const fetchOneCourseFailure = (error) => {
  return {
    type: FETCH_ONE_COURSE_FAILURE,
    payload: error
  }
}
```

```
export const fetchOneCourse = (courseId) => {
  return (dispatch) => {

    dispatch(fetchOneCourseRequest())

    axios.get(`https://coursecube.com/jlc-rest-api/mycourse/${courseId}`)
      .then(response => {
        const course = response.data;
        dispatch(fetchOneCourseSuccess(course))
      })
      .catch(error => {
        dispatch(fetchCourseFailure(error))
      })
  }
}
```

## 5. courseReducer.js

```
import { FETCH_COURSES_REQUEST } from './courseTypes';
import { FETCH_COURSES_SUCCESS } from './courseTypes';
import { FETCH_COURSES_FAILURE } from './courseTypes';

import { FETCH_ONE_COURSE_REQUEST } from './courseTypes';
import { FETCH_ONE_COURSE_SUCCESS } from './courseTypes';
import { FETCH_ONE_COURSE_FAILURE } from './courseTypes';

const allCoursesInitialState = {
  loadingStatus: false,
  courseList: [],
  error: {}
}

export const allCourseReducer = (state = allCoursesInitialState, action) => {
  switch (action.type) {

    case FETCH_COURSES_REQUEST:
      return {
        ...state,
        loadingStatus: true,
      }
    case FETCH_COURSES_SUCCESS:
      return {
        loadingStatus: false,
        courseList: action.payload,
        error: {}
      }
  }
}
```



```
    case FETCH_COURSES_FAILURE:
      return {
        loadingStatus: false,
        courseList: [],
        error: action.payload
      }
    default:
      return state;
  }
}

const oneCourseInitialState = {
  loadingStatus: false,
  oneCourse: {},
  error: {}
}

export const oneCourseReducer = (state = oneCourseInitialState, action) => {
  switch (action.type) {
    case FETCH_ONE_COURSE_REQUEST:
      return {
        ...state,
        loadingStatus: true,
      }
    case FETCH_ONE_COURSE_SUCCESS:
      return {
        loadingStatus: false,
        oneCourse: action.payload,
        error: {}
      }
    case FETCH_ONE_COURSE_FAILURE:
      return {
        loadingStatus: false,
        oneCourse: {},
        error: action.payload
      }
    default:
      return state;
  }
}
```

## 6. mystore.js

```
import { createStore, combineReducers, applyMiddleware } from 'redux';

import logger from 'redux-logger';
import thunk from 'redux-thunk';

import { allCourseReducer } from './courses/courseReducer';
import { oneCourseReducer } from './courses/courseReducer';

const rootReducer = combineReducers({
  mycourses: allCourseReducer,
  mycourse: oneCourseReducer
});

const mystore = createStore(rootReducer, applyMiddleware(logger, thunk));

export default mystore;
```

## 7. App.js

```
import React, { Component } from "react";
import "bootstrap/dist/css/bootstrap.min.css";

import JLCHeader from "./JLCHeader";
import JLCBody from "./JLCBody";

import { Provider } from 'react-redux';
import mystore from './myredux/mystore';
import { BrowserRouter } from "react-router-dom";

class App extends Component {
  render() {
    return (
      <BrowserRouter>
        <Provider store={mystore}>
          <div className="card">
            <JLCHeader />
            <JLCBody />
          </div>
        </Provider>
      </BrowserRouter>
    );
  }
}
export default App;
```



## 8. JLCHheader.js

```
import React, { Component } from "react";
import { Link } from "react-router-dom";

class JLCHheader extends Component {
  render() {
    return (
      <div className="container text-center">
        <h1> Welcome to CourseCube!!!</h1>
        <ul className="nav justify-content-center">
          <li className="nav-item">
            <Link to="/" className="nav-link"> Home </Link>
          </li>
        </ul>
      </div>
    );
  }
}
export default JLCHheader;
```

## 9. JLCBody.js

```
import React, { Component } from "react";
import { Route } from "react-router-dom";

import CourseDetailsComponent from "../CourseDetailsComponent";
import CourseListComponent from "../CourseListComponent";

class JLCBody extends Component {
  render() {
    return (
      <div className="card">
        <Route path="/" component={CourseListComponent} exact />
        <Route path="/mycourse-info/:courseId" component={CourseDetailsComponent} exact />
      </div>
    );
  }
}

export default JLCBody;
```

## 10. CourseListComponent.js

```
import React, { useEffect } from "react";
import { connect } from 'react-redux';
import { Link } from "react-router-dom";

import { fetchCourses } from './myredux/course/courseActions';

const CourseListComponent = ({mycourseData, courseDispatch }) => {

  useEffect(() => {
    courseDispatch(fetchCourses());
  }, []);

  let messageToDisplay = null;
  if (mycourseData.loadingStatus) {
    messageToDisplay = "Loading .....Please Wait.... ";
  } else if (mycourseData.error) {
    messageToDisplay = mycourseData.error.message;
  }

  let coursesToDislay = null;
  if (mycourseData && mycourseData.courseList) {
    coursesToDislay = mycourseData.courseList;
  }

  const mycourseList = coursesToDislay.map(
    mycourse => {
      return (
        <tr key={mycourse.courseId}>
          <td> {mycourse.courseId} </td>
          <td> {mycourse.courseName} </td>
          <td> {mycourse.duration} </td>
          <td> {mycourse.trainer} </td>
          <td> {mycourse.enrollments} </td>
          <td>
            <Link className="btn btn-success" to={"/mycourse-info/" + mycourse.courseId}>
              View Details</Link>
          </td>
        </tr>
      );
    }
  );
};
```



```
return (  
  <div className="container">  
  
    <table className="table">  
      <thead>  
        <tr>  
          <th> Course ID</th>  
          <th> Course Name</th>  
          <th> Duration</th>  
          <th> Trainer</th>  
          <th> Enrollments</th>  
          <th> </th>  
        </tr>  
      </thead>  
      <tbody>  
        {mycourseList}  
      </tbody>  
    </table>  
    <h2 className="text-center"> {messageToDisplay} </h2>  
  
  </div>  
);  
}  
const mapState = state => {  
  return {  
    mycourseData: state.mycourses  
  }  
}  
  
const mapDispatch = dispatch => {  
  return {  
    courseDispatch : dispatch  
  }  
}  
  
export default connect(mapState, mapDispatch)(CourseListComponent);
```





## 11. CourseDetailsComponent.js

```
import React, { useEffect } from "react";
import { connect } from 'react-redux';
import { fetchOneCourse } from './myredux/course/courseActions';

const CourseDetailsComponent = ({ match, mycourseData, courseDispatch }) => {

  useEffect(() => {
    courseDispatch(fetchOneCourse(match.params.courseId));
  }, []);

  let messageToDisplay = null;
  if (mycourseData.loadingStatus) {
    messageToDisplay = "Loading .....Please Wait.... ";
  } else if (mycourseData.error) {
    messageToDisplay = mycourseData.error.message;
  }

  let courseDetails = null;
  if (mycourseData && mycourseData.oneCourse) {
    courseDetails = mycourseData.oneCourse;
  }

  return (
    <div className="container">
      <h2 className="text-center"> {messageToDisplay} </h2>

      <table className="table">
        <tbody>
          <tr>
            <td> Course Id </td>
            <td> {courseDetails.courseId} </td>
            <td> CourseName </td>
            <td> {courseDetails.courseName} </td>
          </tr>
          <tr>
            <td> Trainer </td>
            <td> {courseDetails.trainer} </td>
            <td> Duration </td>
            <td> {courseDetails.duration} </td>
          </tr>
          <tr>
            <td> Enrollments </td>
            <td> {courseDetails.enrollments} </td>
            <td> Price </td>
            <td> Rs.{courseDetails.price} </td>
          </tr>
        </tbody>
      </table>
    </div>
  );
}
```



```
</tr>
<tr>
<td> Discount </td>
<td> {courseDetails.discount}% </td>
<td> Offer Price </td>
<td> Rs.{courseDetails.offerPrice} </td>
</tr>
<tr>
<td>Training Model</td>
<td colspan="3" align="left">
{courseDetails.trainingModel}
</td>
</tr>
<tr>
<td> Who can Learn this? </td>
<td colspan="3" align="left">
{courseDetails.description1}
</td>
</tr>
<tr>
<td> Timings </td>
<td colspan="3" align="left">
{courseDetails.description2}
</td>
</tr>
</tbody>
</table>
</div>
);
}
```

```
const mapState = state => {
  return {
    mycourseData: state.mycourse
  }
}
```

```
const mapDispatch = dispatch => {
  return {
    courseDispatch : dispatch
  }
}
```

```
export default connect(mapState, mapDispatch)(CourseDetailsComponent);
```