Exercise (Instructions): Introduction to Redux

Objectives and Outcomes

In this exercise you will learn to use Redux. You will install and configure Redux and use it within your React application. At the end of this exercise you will be able to:

- Install and configure Redux within your application
- Configure your React application to make use of Redux

Installing and Configuring Redux

- As a first step you will install Redux and React-Redux into your application as follows:

```
yarn add redux@3.7.2
yarn add react-redux@5.0.7
```

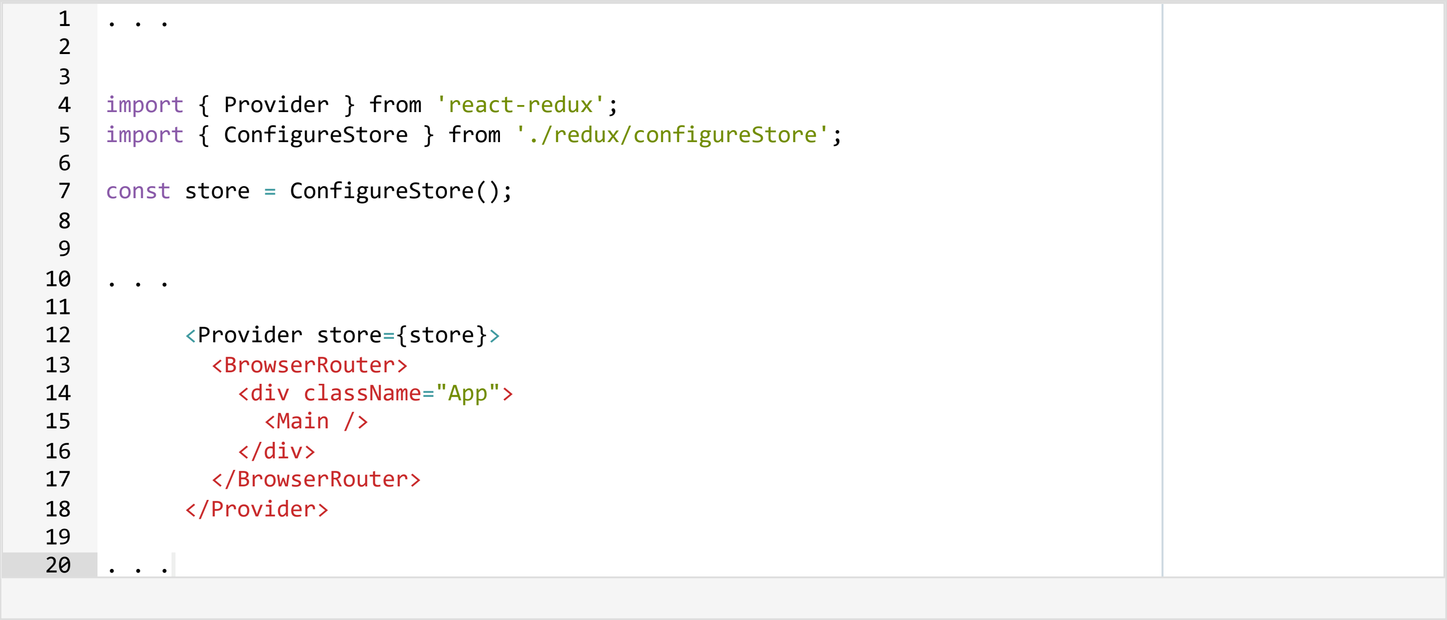
- Next, create a folder named *redux* in the *src* folder and then add a file named *reducer.js* with the code below:

```
import { DISHES } from '../shared/dishes';
import { COMMENTS } from '../shared/comments';
import { PROMOTIONS } from '../shared/promotions';
import { LEADERS } from '../shared/leaders';

export const initialState = {
    dishes: DISHES,
    comments: COMMENTS,
    promotions: PROMOTIONS,
    leaders: LEADERS
};

export const Reducer = (state = initialState, action) => {
    return state;
};
```

- Then, add a file named *configureStore.js* in the *redux* folder and add the following code to it:

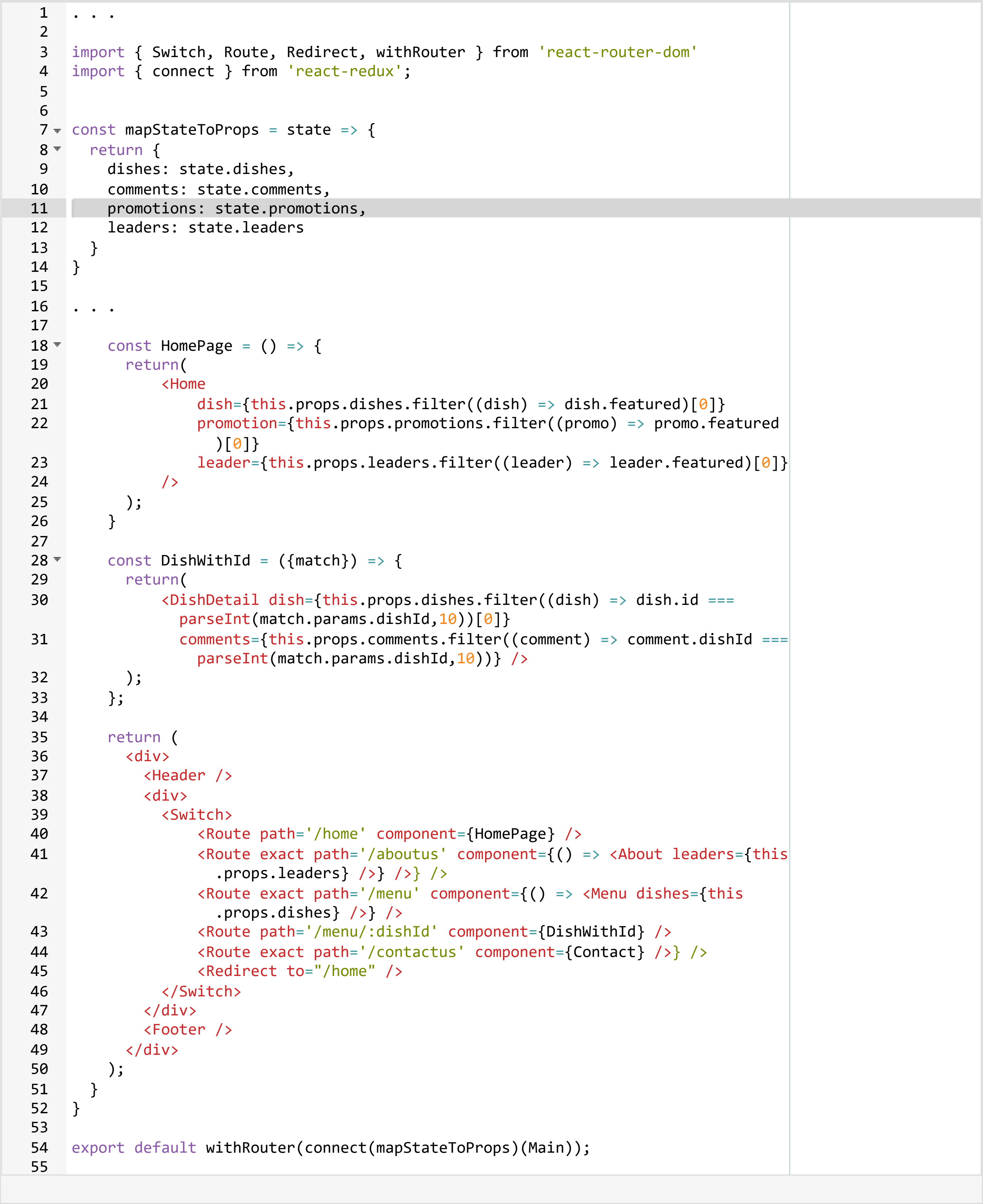
```
import {createStore} from 'redux';
import { Reducer, initialState } from './reducer'

export const ConfigureStore = () => {
    const store = createStore(
        Reducer, // reducer
        initialState, // our initialState
    );

    return store;
}
```

- Next, open *App.js* and update it as follows:



```
. . .


import { Provider } from 'react-redux';
import { ConfigureStore } from './redux/configureStore';

const store = ConfigureStore();


. . .

      <Provider store={store}>
        <BrowserRouter>
          <div className="App">
            <Main />
          </div>
        </BrowserRouter>
      </Provider>

. . .
```

- Finally, update MainComponent.js to connect it to Redux store and use it:



```
. . .

import { Switch, Route, Redirect, withRouter } from 'react-router-dom'
import { connect } from 'react-redux';


const mapStateToProps = state => {
  return {
    dishes: state.dishes,
    comments: state.comments,
    promotions: state.promotions,
    leaders: state.leaders
  }
}

. . .

    const HomePage = () => {
      return(
          <Home
              dish={this.props.dishes.filter((dish) => dish.featured)[0]}
              promotion={this.props.promotions.filter((promo) => promo.featured)[0]}
              leader={this.props.leaders.filter((leader) => leader.featured)[0]}
          />
      );
    }

    const DishWithId = ({match}) => {
      return(
          <DishDetail dish={this.props.dishes.filter((dish) => dish.id === parseInt(match.params.dishId,10))[0]}
            comments={this.props.comments.filter((comment) => comment.dishId === parseInt(match.params.dishId,10))} />
      );
    };

    return (
      <div>
        <Header />
        <div>
          <Switch>
              <Route path='/home' component={HomePage} />
              <Route exact path='/aboutus' component={() => <About leaders={this.props.leaders} />} />} />
              <Route exact path='/menu' component={() => <Menu dishes={this.props.dishes} />} />
              <Route path='/menu/:dishId' component={DishWithId} />
              <Route exact path='/contactus' component={Contact} />} />
              <Redirect to="/home" />
          </Switch>
        </div>
        <Footer />
      </div>
    );
  }
}

export default withRouter(connect(mapStateToProps)(Main));

```

- Save all the changes and do a Git commit with the message "Intro. to Redux".

Conclusions

In this exercise you learnt to install and configure Redux and use it in your React application.