# Exercise (Instructions): Using Redux in your React Native App

## Objectives and Outcomes

In this exercise you will set up the components in your React Native app to connect to Redux and obtain the state information from there. At the end of this exercise you will be able to:

- Set up your React Native app to connect to Redux
- Obtain the state information from Redux
- Dispatch actions to the Redux store to effect changes to the state

## Connecting to Redux

- First, open App.js and update it to enable your React Native app to make use of Redux as follows:

```
. . .

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

const store = ConfigureStore();

. . .

      <Provider store={store}>
        <Main />
const mapStateToProps = state => {
    return {
      leaders: state.leaders
    }
  }

. . .

                    leftAvatar={{source: {uri: baseUrl + item.image}}}

. . .

                <FlatList
                    data={this.props.leaders.leaders}
                    renderItem={renderLeader}
                    keyExtractor={item => item.id.toString()}
                    />

. . .


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

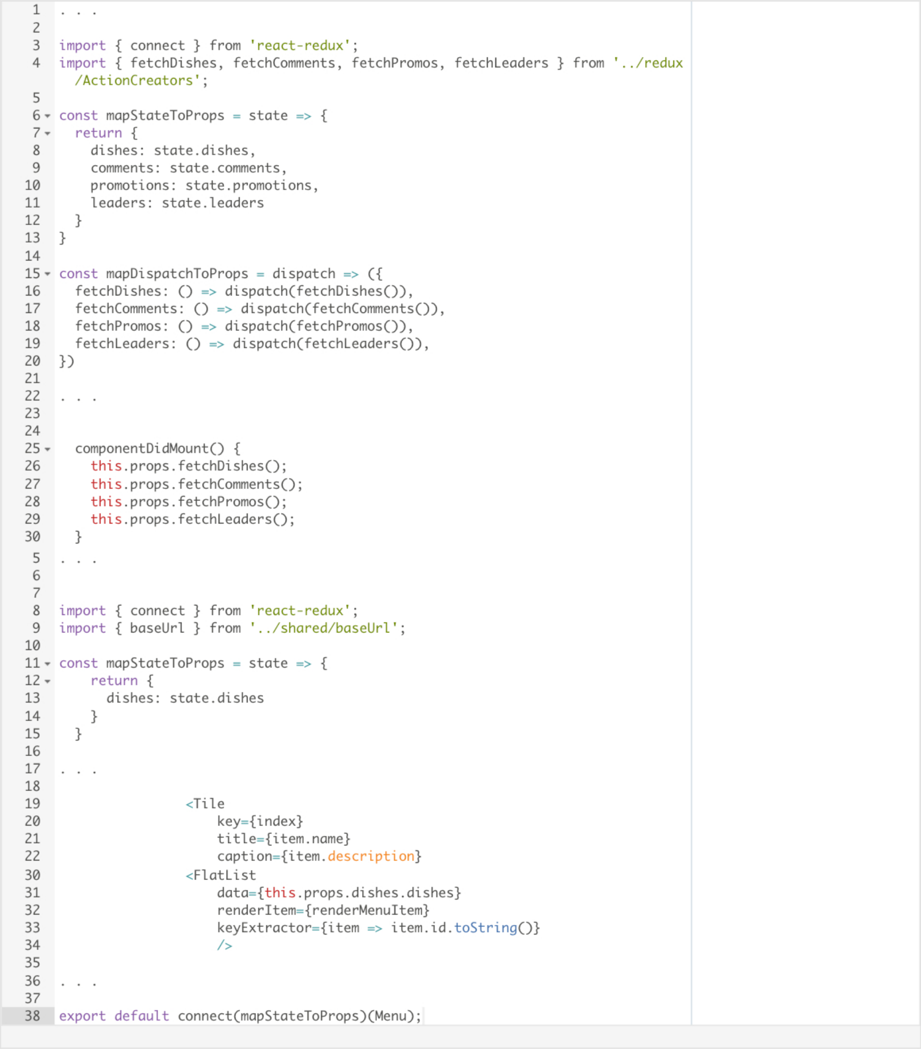
- Then, open DishdetailComponent.js and update it as follows:

```
. . .

import { connect } from 'react-redux';
import { baseUrl } from '../shared/baseUrl';

const mapStateToProps = state => {
    return {
                    favorite={this.state.favorites.some(el => el === dishId)}
                    onPress={() => this.markFavorite(dishId)}
                    />
                <RenderComments comments={this.props.comments.comments.filter
                  ((comment) => comment.dishId === dishId)} />
            </ScrollView>

. . .

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

- Then, open HomeComponent.js and update it as follows:

```
. . .

import { connect } from 'react-redux';
import { baseUrl } from '../shared/baseUrl';

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

. . .

                    image={{uri: baseUrl + item.image}}>

. . .
```

- Then, open MainComponent.js and update it as follows:



```
. . .

import { connect } from 'react-redux';
import { fetchDishes, fetchComments, fetchPromos, fetchLeaders } from '../redux/ActionCreators';

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

const mapDispatchToProps = dispatch => ({
  fetchDishes: () => dispatch(fetchDishes()),
  fetchComments: () => dispatch(fetchComments()),
  fetchPromos: () => dispatch(fetchPromos()),
  fetchLeaders: () => dispatch(fetchLeaders()),
})

. . .


  componentDidMount() {
    this.props.fetchDishes();
    this.props.fetchComments();
    this.props.fetchPromos();
    this.props.fetchLeaders();
  }
. . .


import { connect } from 'react-redux';
import { baseUrl } from '../shared/baseUrl';

const mapStateToProps = state => {
    return {
      dishes: state.dishes
    }
  }

. . .

                <Tile
                    key={index}
                    title={item.name}
                    caption={item.description}
                <FlatList
                    data={this.props.dishes.dishes}
                    renderItem={renderMenuItem}
                    keyExtractor={item => item.id.toString()}
                    />

. . .

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

- Save all the changes and do a Git commit with the message "Using Redux"

## Conclusions

In this exercise you have updated your React app to connect to Redux