

## Exercise (Instructions): Setting up Redux

### Objectives and Outcomes

In this exercise you will set up Redux in your React Native application. At the end of this exercise you will be able to:

- Set up Redux for your app
- Create reducers, actions and action creators

### Setting up Redux

- First, set up Redux and allied packages by typing the following at the prompt in your project folder:

```
1 yarn add redux@4.0.0
2 yarn add react-redux@5.0.7
3 yarn add redux-thunk@2.2.0
4 yarn add redux-logger@3.0.6
```

- Find out the IP address of your computer. On a Mac, go to System Preferences->Network to find the IP address of your network adapter. If you are using a Microsoft Windows based machine, you can use the instructions available at <http://windows.microsoft.com/en-us/windows/find-computers-ip-address#1TC=windows-7> to find your machine's IP address.
- Next, create a file named baseUrl.js in the shared folder and add the following to it:

```
1 export const baseUrl = 'http://<Your Computer\'s IP address>:3001/'
```

- For example, if your IP address is 192.168.1.121, then the above line should be as follows:

```
1 export const DISHES_LOADING = 'DISHES_LOADING';
2 export const ADD_DISHES = 'ADD_DISHES';
3 export const DISHES_FAILED = 'DISHES_FAILED';
4 export const ADD_COMMENTS = 'ADD_COMMENTS';
5 export const COMMENTS_FAILED = 'COMMENTS_FAILED';
6 export const PROMOS_LOADING = 'PROMOS_LOADING';
7 export const ADD_PROMOS = 'ADD_PROMOS';
8 export const PROMOS_FAILED = 'PROMOS_FAILED';
9 export const LEADERS_LOADING = 'LEADERS_LOADING';
10 export const ADD_LEADERS = 'ADD_LEADERS';
11 export const LEADERS_FAILED = 'LEADERS_FAILED';
```

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

```
1 import {createStore, combineReducers, applyMiddleware} from 'redux';
2 import thunk from 'redux-thunk';
3 import logger from 'redux-logger';
4 import { dishes } from './dishes';
5 import { comments } from './comments';
6 import { promotions } from './promotions';
7 import { leaders } from './leaders';
8
9 export const ConfigureStore = () => {
10   const store = createStore(
11     combineReducers({
12       dishes,
13       comments,
14       promotions,
15       leaders
16     }),
17     applyMiddleware(thunk, logger)
18   );
19   return store;
20 }
21
22 // Initial state
23 dishes: [], action) => {
24   switch (action.type) {
25     case ActionTypes.ADD_DISHES:
26       return {...state, isLoading: false, errMsg: null, dishes: action
27         .payload};
28     case ActionTypes.DISHES_LOADING:
29       return {...state, isLoading: true, errMsg: null, dishes: []}
30     case ActionTypes.DISHES_FAILED:
31       return {...state, isLoading: false, errMsg: action.payload};
32     default:
33       return state;
34   }
35 }
36 };
```

- Then, add a file named comments.js in the redux folder and update it as follows:

```
1 import * as ActionTypes from './ActionTypes';
2
3 export const comments = (state = { errMsg: null, comments: [] }, action) => {
4   switch (action.type) {
5     case ActionTypes.ADD_COMMENTS:
6       return {...state, errMsg: null, comments: action.payload};
7     case ActionTypes.COMMENTS_FAILED:
8       return {...state, errMsg: action.payload};
9     default:
10       return state;
11   }
12 }
13 };
```

```
1 case ActionTypes.ADD_LEADERS:
2   return {...state, isLoading: false, errMsg: null, leaders: action
3     payload};
```

```

9
10 case ActionTypes.LEADERS_LOADING:
11   return {...state, isLoading: true, errMsg: null, leaders: []}
12
13 case ActionTypes.LEADERS_FAILED:
14   return {...state, isLoading: false, errMsg: action.payload};
15
16 default:
17   return state;
18 }
19 };

```

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

```

1 import * as ActionTypes from './ActionTypes';
2
3 export const promotions = (state = { isLoading: true,
4   errMsg: null,
5   promotions: [], action } => {
6   switch (action.type) {
7     case ActionTypes.ADD_PROMOS:
8       return {...state, isLoading: false, errMsg: null, promotions: action
9         .payload};
10
11     case ActionTypes.PROMOS_LOADING:
12       return {...state, isLoading: true, errMsg: null, promotions: []}
13
14     case ActionTypes.PROMOS_FAILED:
15       return {...state, isLoading: false, errMsg: action.payload};
16
17     default:
18       return state;
19   }
20 }
21
22 export const fetchComments = () => (dispatch) => {
23   return fetch(baseUrl + 'comments')
24     .then(response => {
25       if (response.ok) {
26         return response;
27       } else {
28         var error = new Error('Error ' + response.status + ': ' + response
29           .statusText);
30         error.response = response;
31         throw error;
32       }
33     },
34     error => {
35       var errmsg = new Error(error.message);
36       throw errmsg;
37     })
38     .then(response => response.json())
39     .then(comments => dispatch(addComments(comments)))
40     .catch(error => dispatch(commentsFailed(error.message)));
41 };
42
43 export const commentsFailed = (errmsg) => ({
44   type: ActionTypes.COMMENTS_FAILED,
45   payload: errmsg
46 });
47
48 export const addComments = (comments) => ({
49   type: ActionTypes.ADD_COMMENTS,
50   payload: comments
51 });
52
53 export const fetchDishes = () => (dispatch) => {
54   dispatch(dishesLoading());
55
56   return fetch(baseUrl + 'dishes')
57     .then(response => {
58       if (response.ok) {
59         return response;
60       } else {
61         var error = new Error('Error ' + response.status + ': ' + response
62           .statusText);
63         error.response = response;
64         throw error;
65       }
66     },
67     error => {
68       var errmsg = new Error(error.message);
69       throw errmsg;
70     })
71     .then(response => response.json())
72     .then(dishes => dispatch(addDishes(dishes)))
73     .catch(error => dispatch(dishesFailed(error.message)));
74 };
75
76 export const dishesLoading = () => ({
77   type: ActionTypes.DISHES_LOADING
78 });
79
80 export const dishesFailed = (errmsg) => ({
81   type: ActionTypes.DISHES_FAILED,
82   payload: errmsg
83 });
84
85 export const addDishes = (dishes) => ({
86   type: ActionTypes.ADD_DISHES,
87   payload: dishes
88 });
89
90 export const fetchPromos = () => (dispatch) => {
91   dispatch(promosLoading());
92
93   return fetch(baseUrl + 'promotions')
94     .then(response => {
95       if (response.ok) {
96         return response;
97       } else {
98         var error = new Error('Error ' + response.status + ': ' + response
99           .statusText);
100         error.response = response;
101         throw error;
102       }
103     },
104     error => {
105       var errmsg = new Error(error.message);
106       throw errmsg;
107     })
108     .then(response => response.json())
109     .then(promos => dispatch(addPromos(promos)))
110     .catch(error => dispatch(promosFailed(error.message)));
111 };
112
113 export const promosFailed = (errmsg) => ({
114   type: ActionTypes.PROMOS_FAILED,
115   payload: errmsg
116 });
117
118 export const addPromos = (promos) => ({
119   type: ActionTypes.ADD_PROMOS,
120   payload: promos
121 });
122
123 export const fetchLeaders = () => (dispatch) => {
124   dispatch(leadersLoading());

```

```

111
112     return fetch(baseUrl + 'leaders')
113     .then(response => {
114       if (response.ok) {
115         return response;
116       } else {
117         var error = new Error('Error ' + response.status + ': ' + response
118           .statusText);
119         error.response = response;
120         throw error;
121       }
122     }, error => {
123     },
124   );
125
126   // Export the actions
127   export const leadersLoading = () => ({
128     type: ActionTypes.LEADERS_LOADING
129   });
130
131   export const leadersFailed = (errmess) => ({
132     type: ActionTypes.LEADERS_FAILED,
133     payload: errmess
134   });
135
136   export const addLeaders = (leaders) => ({
137     type: ActionTypes.ADD_LEADERS,
138     payload: leaders
139   });
140
141   // Export the actions
142   export const addLeaders = (leaders) => ({
143     type: ActionTypes.ADD_LEADERS,
144     payload: leaders
145   });

```

- Save all the changes and do a Git commit with the message "Setting up Redux".

## Conclusions

In this exercise you have set up Redux support within your React Native app.

Mark as completed