

Exercise (Instructions): Combining Reducers

Objectives and Outcomes

In this exercise you will learn about how we can have separate reducers managing parts of the state, and how they can be combined together to manage the whole state. At the end of this exercise you will be able to:

- Implement reducers that are responsible for only part of the state
- Combine the reducers to manage the entire state

Splitting the Reducer

- In the redux folder, create a new file named dishes.js and add the following to it:

```
1 import { DISHES } from '../shared/dishes';
2
3 export const Dishes = (state = DISHES, action) => {
4   switch (action.type) {
5     default:
6       return state;
7   }
8 };
```

```
3 export const Promotions = (state = PROMOTIONS, action) => {
4   switch (action.type) {
5     default:
6       return state;
7   }
8 };
```

- And finally, create a new file named leaders.js and add the following to it:

```
1 import { LEADERS } from '../shared/leaders';
2
3 export const Leaders = (state = LEADERS, action) => {
4   switch (action.type) {
5     default:
6       return state;
7   }
8 };
9
10 export const configureStore = () => {
11   const store = createStore(
12     combineReducers({
13       dishes: Dishes,
14       comments: Comments,
15       promotions: Promotions,
16       leaders: Leaders
17     })
18   );
19   return store;
20 };
```

- Now we can safely delete the reducer.js file from the project.
- Save all the changes and do a Git commit with the message "Combining Reducers".

Conclusions

In this exercise we have learnt to split the reducer into multiple reducers that manage partial state, and then combine them together.