



Welcome 2 REACT 4 (Redux)

by Toon Nelissen & David Singh (AppSaloon)



Lesson's 1,2 & 3

- JSX
- Rendering an element
- React development tools
- Component & props
- Create react app
- List & keys
- State & Lifecycle
- Events
- Forms
- Lifting state

Redux

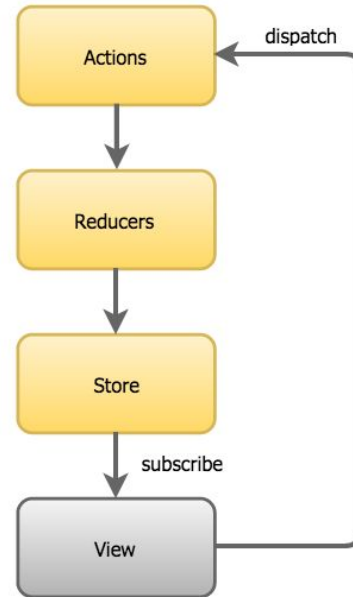


Why do we need Redux?

Because things get to complicated

Redux?

- Single source of truth
STORE
- STATE is read-only
ACTIONS
- Changes are made with pure functions
REDUCERS





ACTIONS



Actions

Actions are payloads of information that send data from your application to your store.

```
1  function addTodo(text) {  
2    return {  
3      type: ADD_TODO,  
4      text  
5    }  
6  }
```

REDUCERS



Reducers

Reducers specify how the application's state changes in response to actions sent to the store. Remember that actions only describe what happened, but don't describe how the application's state changes.

```
10 function todoApp(state = initialState, action) {
11   switch (action.type) {
12     case SET_VISIBILITY_FILTER:
13       return Object.assign({}, state, {
14         visibilityFilter: action.filter
15       })
16     case ADD_TODO:
17       return Object.assign({}, state, {
18         todos: [
19           ...state.todos,
20           {
21             text: action.text,
22             completed: false
23           }
24         ]
25       })
26     default:
27       return state
28   }
29 }
```

—

STORE



Store

The **Store** is the object that brings them together. The store has the following responsibilities:

- Holds application state;
- Allows access to state via `getState()`;
- Allows state to be updated via `dispatch(action)`;
- Registers listeners via `subscribe(listener)`;
- Handles unregistering of listeners via the function returned by `subscribe(listener)`.

```
1 import { createStore } from 'redux'
2 import todoApp from './reducers'
3
4 const store = createStore(todoApp)
```