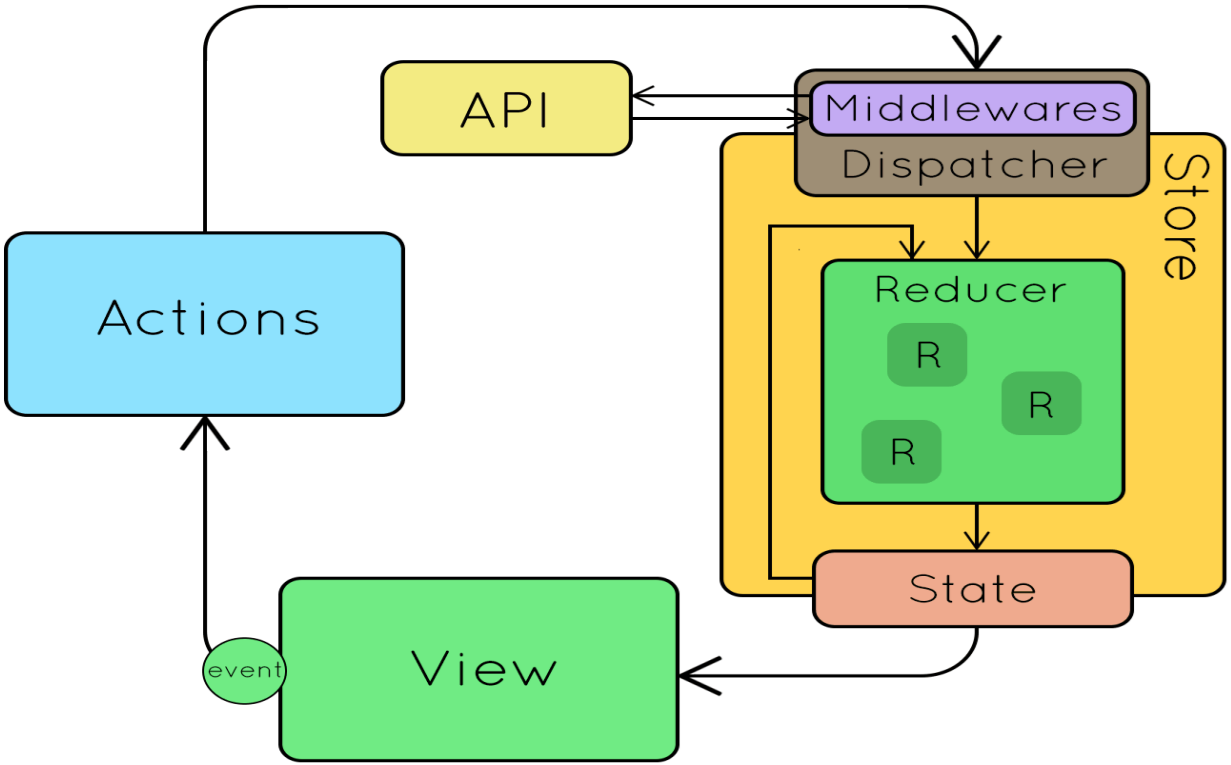
Redux (4 Units)

Unit 5: Why redux


API
Middlewares
Dispatcher
Store
Actions
Reducer
R
R
R
State
event
View

Overview

Single source of truth

The state of your whole application is stored in an object tree within a single store.

Overview

```
console.log(store.getState())

/* Prints
{
  todos: [
    {
      text: 'Consider using Redux',
      completed: true,
    },
    {
      text: 'Keep all state in a single tree',
      completed: false
    }
  ]
}
*/
```

Overview

State is read-only

The only way to change the state is to emit an action, an object describing what happened.

Overview

```
store.dispatch({
  type: 'COMPLETE_TODO',
  index: 1
})

store.dispatch({
  type: 'SET_VISIBILITY_FILTER',
  filter: 'SHOW_COMPLETED'
})
```

Overview

Changes are made with pure functions

To specify how the state tree is transformed by actions, you write pure reducers.

Overview

```
function todos(state = [], action) {
  switch (action.type) {
    case 'ADD_TODO':
      return [
        ...state,
        {
          text: action.text,
          completed: false
        }
      ]

    default:
      return state
  }
}

let store = createStore(todos)
```