



## **Redux - State container for JS apps**

Miguel Braga - Front-end Developer

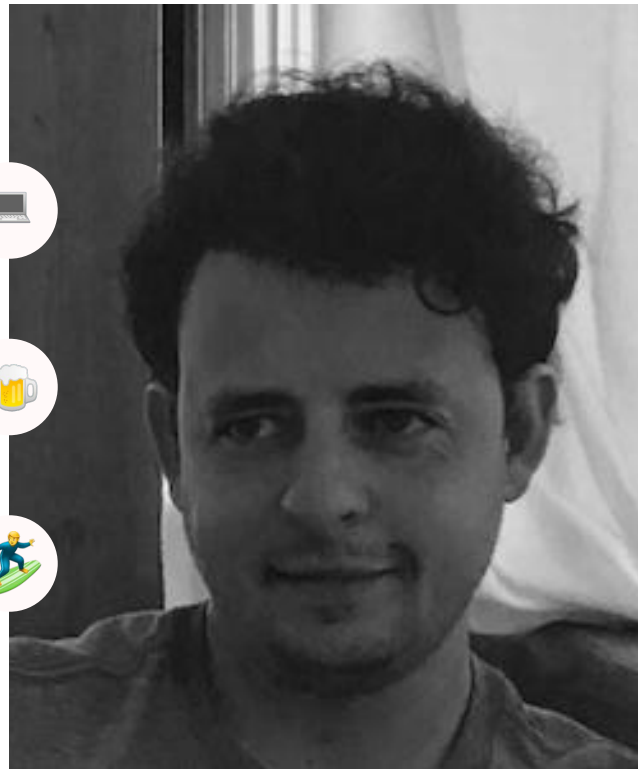
# What to expect?

1. To-Do list with vanilla JavaScript
2. To-Do list with vanilla JavaScript and Redux
3. To-Do list with React and Redux
4. To-Do list with React, Redux and calling an external API

# Miguel Braga

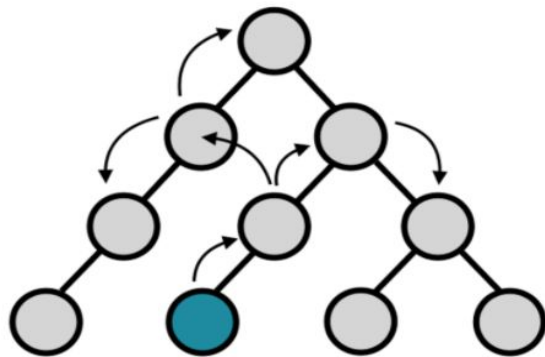
Web Development Lead Teacher

**About me:** I've been working as a software engineer since 2006, developed apps in Portugal, Ireland, Spain and Denmark. I'm currently working as a front-end developer for a music startup and as a lead teacher at Ironhack.

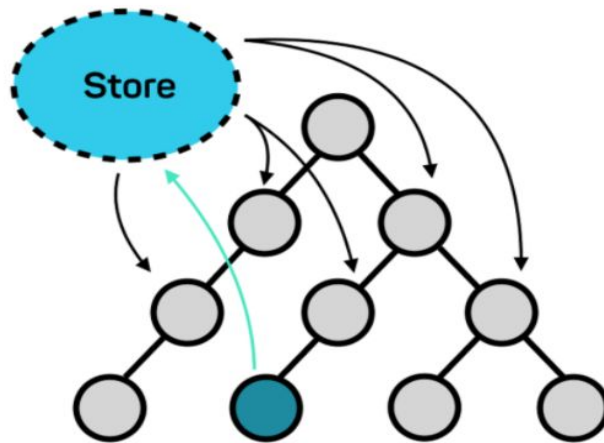


# Managing state

Without Redux



With Redux



Component initiating change

# Redux

## Predictable state container



### Predictable

Redux helps you write applications that **behave consistently**, run in different environments (client, server, and native), and are **easy to test**.



### Centralized

Centralizing your application's state and logic enables powerful capabilities like **undo/redo**, **state persistence**, and much more.



### Debuggable

The Redux DevTools make it easy to trace **when, where, why, and how your application's state changed**. Redux's architecture lets you log changes, use **"time-travel debugging"**, and even send complete error reports to a server.



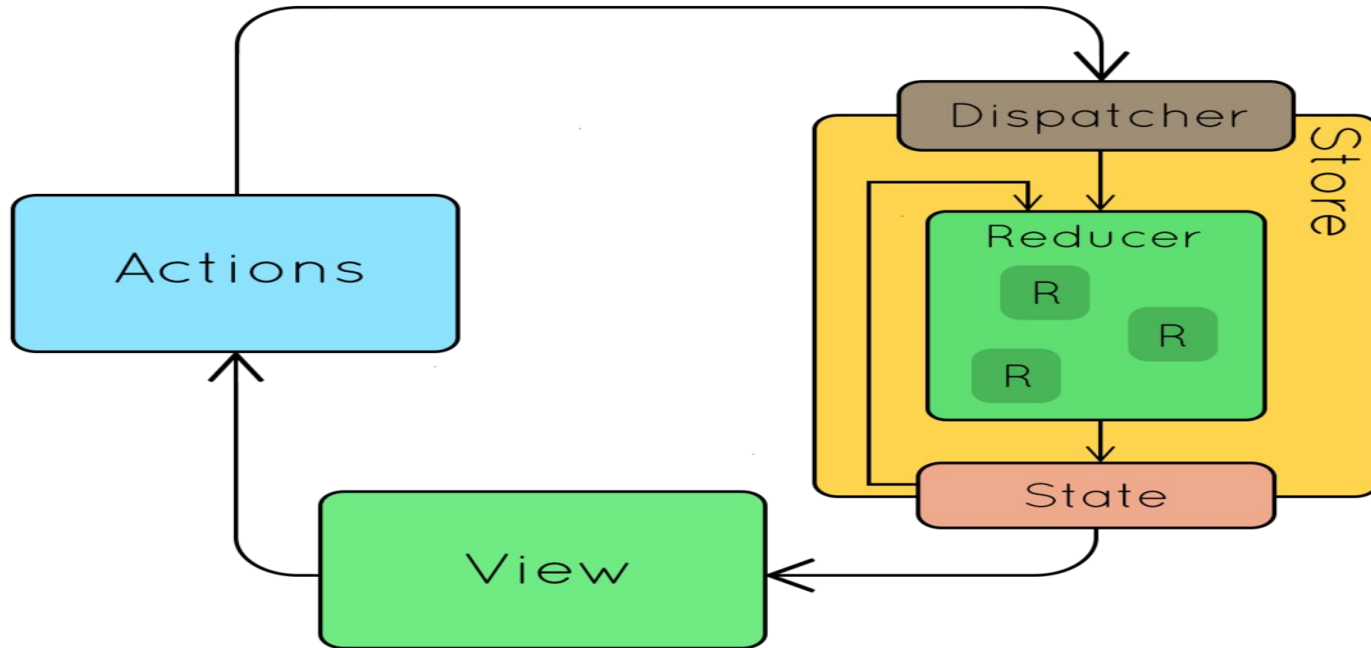
### Flexible

Redux **works with any UI layer**, and has **a large ecosystem of addons** to fit your needs.

## Three Redux principles

1. **Single source of truth** - The global state of your application is stored in an object tree within a single store.
2. **State is read-only** - The only way to change the state is to emit an action, an object describing what happened.
3. **Changes are made with immutable functions** - To specify how the state is transformed you write immutable reducers.

# Redux flow



# Lab

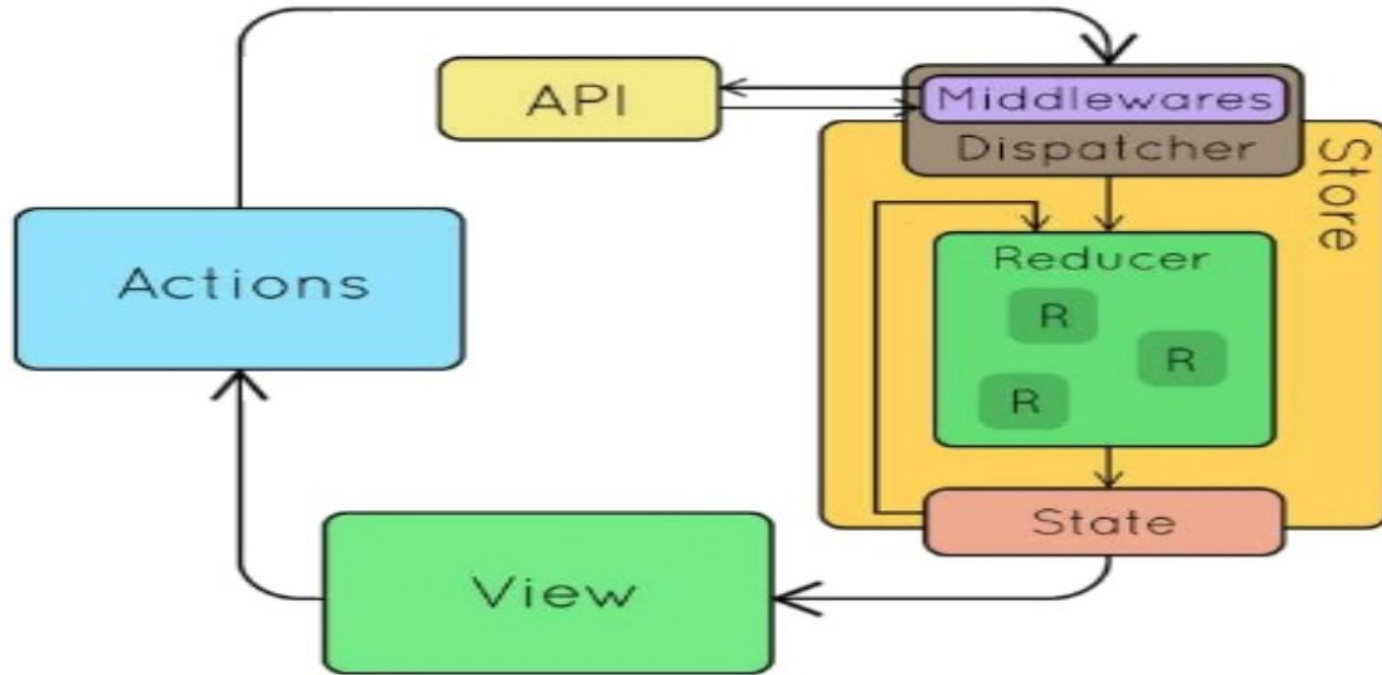
## **Toggle todo action**

Whenever you click a todo it should update a todo property called “completed”.

Depending on the todo “completed” value, the todo title should be strikethrough.



## External API - Updating the store flow



# Redux-Thunk

**Redux Thunk** is a middleware that lets you call action creators that return a function instead of an action object.

## Lab

### **Get the logged user by using an endpoint**

1. Create the proper API call on api.js.
2. Dispatch the action on the loggedUser.js (actions folder).
3. Use mapDispatchToProps on app.js to call the action method.
4. Update the reducer with a new action “SET\_LOGGED\_USER”.