

STATE MANAGEMENT

Jonathan Cheng & Sean Prendi

WHAT IS "STATE MANAGEMENT"?

Recall React components (usually) have state.

State Management (our definition): The practice of "handling" pieces of state in a productive way.

Handling = passing, receiving, modifying, saving, fetching, pushing, waving it in the air (like you just don't care)

Do we just figure it out as we go?

Remember the waterfall model of React state! Not as easy an endeavor as it may seem at first.

WHY STATE MANAGEMENT?

Sometimes state is simple: { name = “Mr. Meeseeks” }

Sometimes it's not as simple:

```
{
```

```
  numMeeseeks = 5,
```

```
  spawnMeeseeks = (name) => { /* code */ },
```

```
  mission = (goal) => { /* code */ },
```

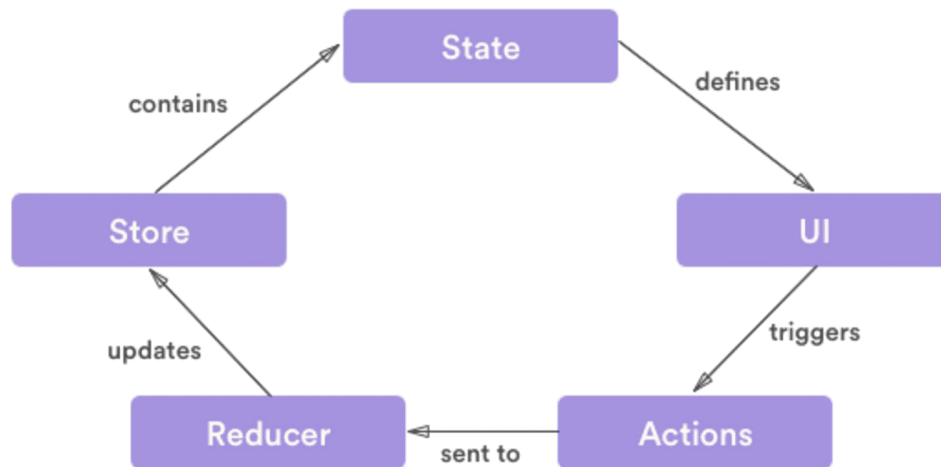
```
  despawnMeeseeks = () => { /* some callback */ }
```

```
}
```

THE OLD WAY

A Predictable State Container for JS Apps: REDUX

Not actually packaged/connected to React, though they are often paired together (very often).



ACTIONS (REDUX.JS.ORG)

- Payloads of information
- Application --(data)-> action

```
{  
  type: "ADD_TODO",  
  payload: {text:"Hello Foo"}  
}
```

- “Action creator” = function that takes arguments and returns an action object (as above)

REDUCERS (REDUX.JS.ORG)

- How does state change based on an action?
- App --(data)--> action --> store --> new state

```
const TODOReducer= (state = {}, action) =>
{
  switch (action.type) {
    case "ADD_TODO":
      return {
        ...state,
        ...action.payload
      };
    default:
      return state;
  }
};
```

STORE (REDUX.JS.ORG)

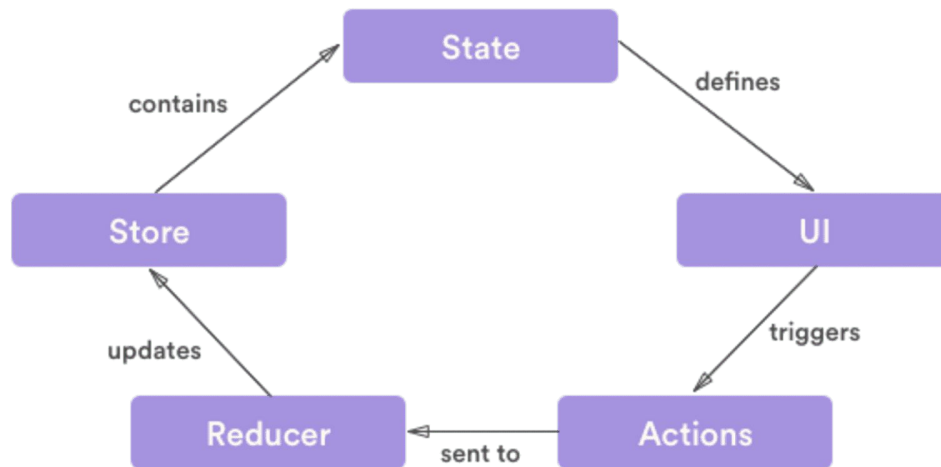
- Simply stores the application state
- Looks like a JS object (or JSON)

```
{  
  A: val1,  
  B: val2,  
  ...  
}
```

THE OLD WAY

A Predictable State Container for JS Apps: REDUX

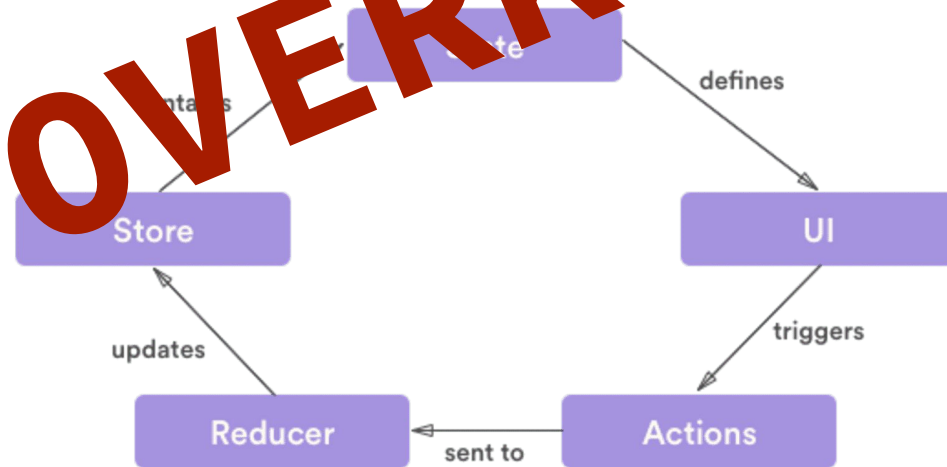
Not actually packaged/connected to React, though they are often paired together (very often).



THE OLD WAY

A Predictable State Container for JS Apps: REDUX

Not actually packaged/connected to React though they are often paired together (very often).



USEFUL, BUT

Not for us mere mortals.

For Facebook, maybe...

Redux is notoriously complicated to learn. Often it is unjustifiable to spend time+effort to use it for a startup-level (low-medium complexity) webapp.

INSTEAD: EASYPEASY

ABSTRACTION of Redux!

- + Maintains the solid principles + concepts
- + No messy set up of action types, reducers, etc.
- + Very little boilerplate (esp. vs Redux)
- Another layer of abstraction
- 3rd party (non-official React/Redux)
- Smaller community

LINKS:

Official Site:

<https://easy-peasy.now.sh/>

- They have great Quick Start docs.
 - Follow those and you're golden!
 -

Neat little article: (also has good step-by-step)

<https://flaviocopes.com/react-easy-peasy/>