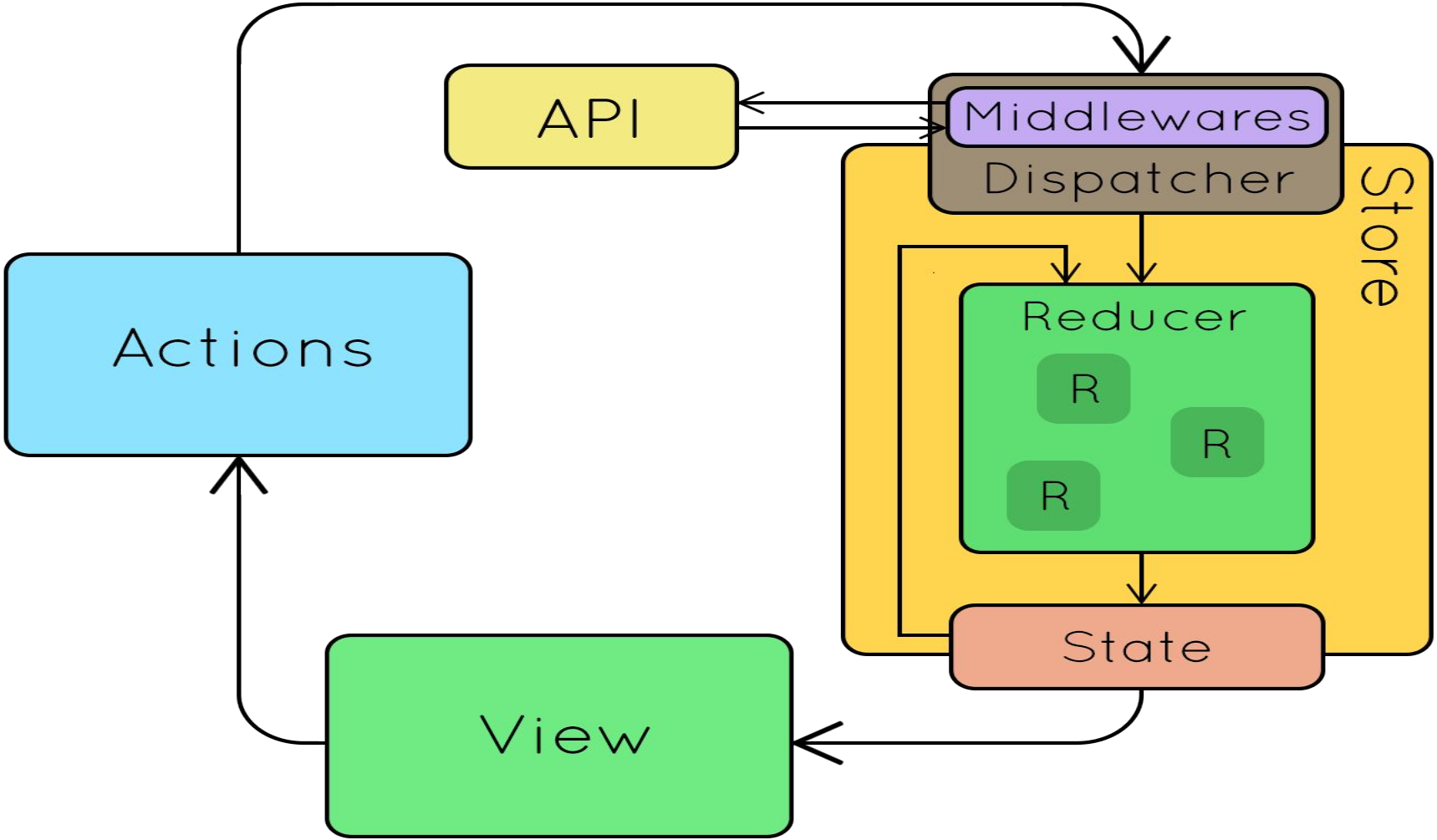
React and Redux: Lesson 5

Using React and Redux

William Abboud

* Covers React v15 - 16


API
Middlewares
Dispatcher
Store
Reducer
R
R
R
State
Actions
View

Redux Data Flow - Fetching Data

In a Redux architecture typically you:

1. Fetch your data from an API
2. Dispatch an Action that you are storing the new data
3. Store that data in the Store's State
4. View gets updated via store.subscribe() automatically


Context

React Context API

Docs: https://reactjs.org/docs/legacy-context.html
React 16.3 comes with a new API but we will be focusing on the old one.
Context provides a way to pass data through the component tree without having to pass props down manually at every level.

Typically you pass data via props. But sometimes you have data that is very general for your whole application such as a theme, a language and so one.

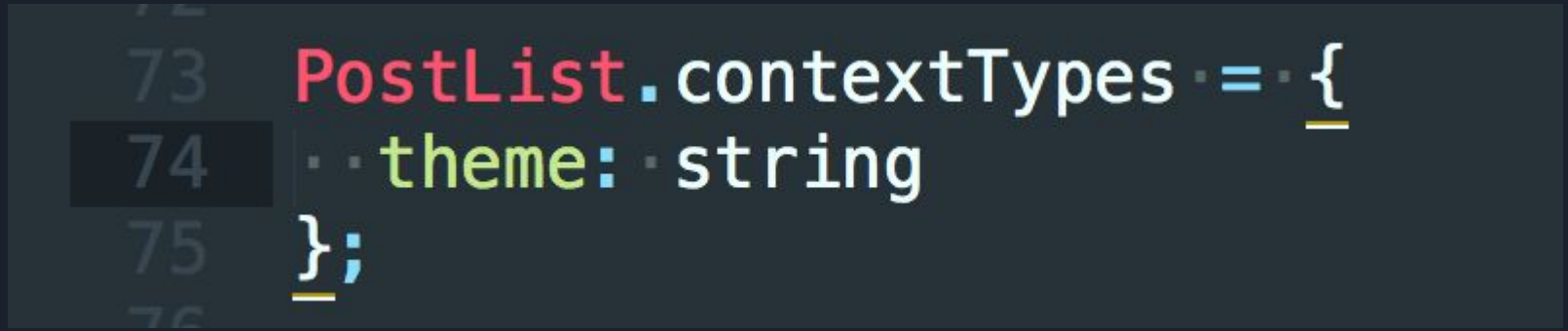
React provides the Context API which is used to plug in data directly into a Component without using props.

getChildContext - is a special method which you use to return the context object, an object which can be accessed via this.context in a child account. You need to supply this method and return an object from it in order to use this.context

Component.childContextTypes - is an object which defines the type of your context properties. You MUST supply it in order to receive the context via this.context.

```
44  App.childContextTypes = {
45    theme: string
46  };
```

Component.contextTypes - is an object which defines the type of your context properties. You MUST define it in order to receive the context via this.context.



React-Redux

React-redux is a very small supplementary library that exposes 2 important parts.

1. connect() function HOC - the connect() function is a High Order Component which returns a connected component - a component which has access to the Store's state, and other Store functionalities.
2. Provider component - in the old context API the provider component is just a wrapper component which inserts the redux Store in the context object.

React-Redux utilities and terms

mapStateToProps - is a function that you pass in to **connect()** which gets the current **state** as a parameter and returns props to component using it. It maps the **Store's state** to the **connected component's props.**

mapDispatchToProps - is a function that you pass in to **connect()** which gets **dispatch** method as an argument and you can use to make function **props** to dispatch **actions**.

Reducers Composition - reducer composition is a pattern where you call reducers from within a main reducer passing them only the state of the app they care about and the current action dispatched.