

REACT AND FRIENDS

- James Kent
- Matt
Gilbride

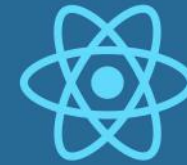
AGENDA

- What is React?
- Intro to Create React App
- What React isn't
- Common companion libraries
- What makes React a good choice for your team

WHAT IS REACT?

- A view library for building component-based UIs
- Created by Facebook in 2013
- Includes a declarative DSL for describing components called JSX
 - "html in your javascript"
- Components are a function of two parameters
 - "props" = read-only
 - "state" = read/write
- Components have a specific lifecycle that can be used to influence behavior
 - <http://projects.wojtekmaj.pl/react-lifecycle-methods-diagram/>

CREATE REACT APP



- <https://github.com/facebook/create-react-app>

WHAT REACT ISN'T

- A "framework"
- The following is up to you:
 - State Management
 - Routing
 - Forms
 - Styling
 - Testing

OUR STACK

STATE MANAGEMENT



Flux



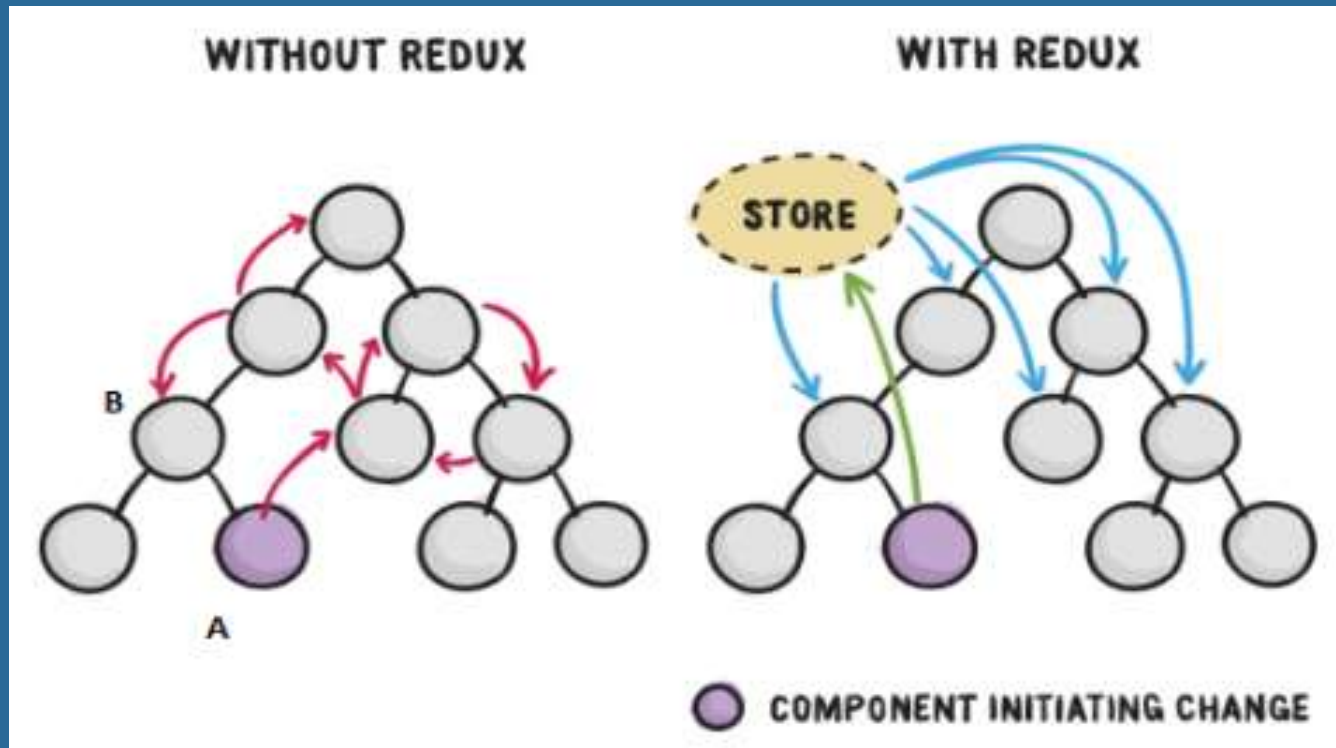
Redux



Mobx

- It is totally possible to use React without any state management library!

REDUX - DATA FLOW



Source: <https://www.slideshare.net/binhqdgmail/006-react-redux-framework>

REDUX - VOCABULARY

- STORE - a single, global data repository and an API to change said data
- STATE - the data
 - Important: this is NOT the same as the "state" of a component : - (
- ACTION - an event
 - Usually, causes some update to STATE
- REDUCER - function which responds to ACTION(s) and decides how to update STATE
- DISPATCH - function that passes actions to reducers
 - you "dispatch" an "action"

REACT-REDUX

- (yet another) a companion library
- provides a nice API for using redux with your React components
- subscribe to pieces of the data, receiving updates when they change
- make changes to the data from your components

REACT-REDUX IN PRACTICE

```
9 interface StateProps {
10   inFlight: boolean;
11   schedule?: Session[];
12 }
13
14 interface DispatchProps {
15   scheduleRequest: () => void;
16 }
17
18 type ScheduleProps = StateProps & DispatchProps;
19
20 class ScheduleComponent extends Component<ScheduleProps> {
21
22   public componentDidMount() {
23     this.props.scheduleRequest();
24   }
25
26   public render() {
27     return (
28       <Fragment>
29         <h3>Choose an appointment time:</h3>
30
31         <div className={styles.schedule}>
32           {(this.props.inFlight || !this.props.schedule)
33             ? <SpinnerComponent />
34             : <Fragment>
35               {this.props.schedule.map((session: Session) => (
36                 <SessionLink
37                   key={session.id}
38                   {...session}
39                 ))}
40             </Fragment>
41         </div>
42       </Fragment>
43     );
44   }
45 }
46
47
48
49 const mapStateToProps: MapStateToProps<StateProps, ScheduleProps, { spa: SpaState }> = state => ({
50   inFlight: state.spa.inFlight,
51   schedule: state.spa.schedule,
52 });
53
54 const mapDispatchToProps: MapDispatchToProps<DispatchProps, ScheduleProps> = dispatch => ({
55   scheduleRequest: () => dispatch(ActionCreators.scheduleRequest()),
56 });
57
58 export const ScheduleContainer = connect(mapStateToProps, mapDispatchToProps)(ScheduleComponent);
59
```

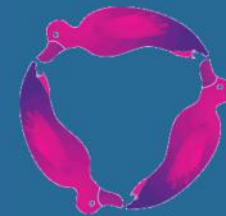
REACT-REDUX IN PRACTICE

- tooling matters

ASYNC ACTIONS



redux-thunk



redux-saga

redux-observable

```
function makeASandwichWithSecretSauce (forPerson) {  
  return function (dispatch) {  
    return fetchSecretSauce().then(  
      sauce => dispatch(makeASandwich(forPerson, sauce)),  
      error => dispatch(apologize('The Sandwich Shop', forPerson, error))  
    );  
  };  
};
```

- a.k.a. but I need to talk to a server!

REDUX-OBSERVABLE

- turns redux actions into observables
- "Epics": actions in, actions out

RXJS



- "Observable" - think of it as a stream
- resources
 - <http://rxmarbles.com/>
 - <http://reactivex.io/documentation/operators/merge.html>
 - <http://reactivex.io/documentation/operators/zip.html>
 - Video: Marble diagrams in Rx

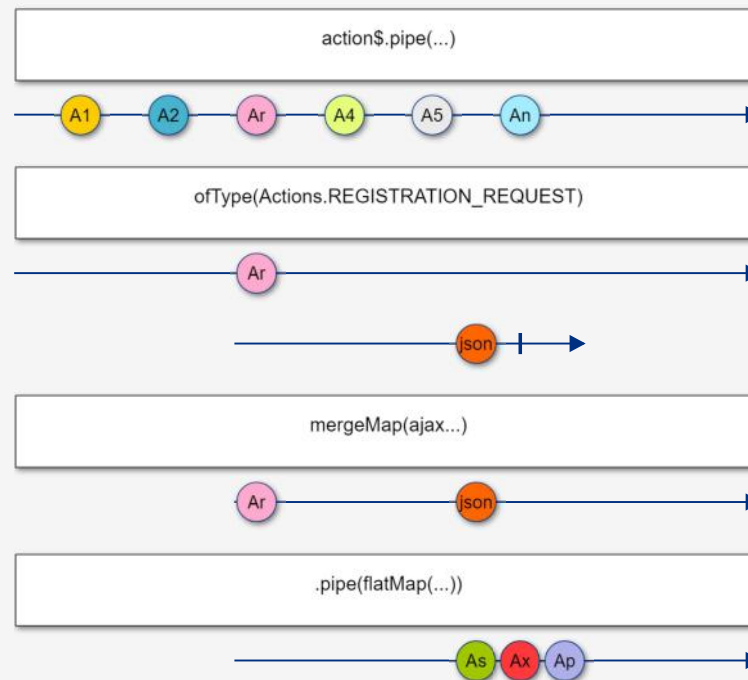
REDUX-OBSERVABLE IN PRACTICE

The screenshot shows a web application for 'Chariot SPA Day' and its Redux DevTools interface. The web application has a dark blue header with the 'Chariot SOLUTIONS' logo. The main content area is white and contains a 'Welcome to the SPA' section with several paragraphs of text. A 'Success! Your registration is complete' message is displayed, followed by appointment details: 'Appointment: Nov. 28th - 10 am - 11 am', 'Name: Matthew T Gilbride', 'Email: mtg5014@gmail.com', and 'Treatment: Chemical peel'. Below this is a 'Complete Registrant List' table with five rows, each showing 'Matthew T Gilbride' with the email 'mtg5014@gmail.com' and the treatment 'Chemical peel'. A 'Back to Schedule' button is at the bottom left, and a green 'CHAT' button is at the bottom right. The Redux DevTools interface on the right shows a list of actions: '@@INIT', 'spa/register/REQUEST', 'spa/register/SUCCESS', 'spa/schedule/REQUEST', '@@router/LOCATION_CHAN...', and 'spa/schedule/SUCCESS'. The 'Diff' view shows a change in the 'inflight' state from 'true' to 'false'.

Name	Email	Treatment
Matthew T Gilbride	mtg5014@gmail.com	Exfoliation
Matthew T Gilbride	mtg5014@gmail.com	Exfoliation
Matthew T Gilbride	mtg5014@gmail.com	Exfoliation
Matthew T Gilbride	mtg5014@gmail.com	Chemical peel
Matthew T Gilbride	mtg5014@gmail.com	Chemical peel

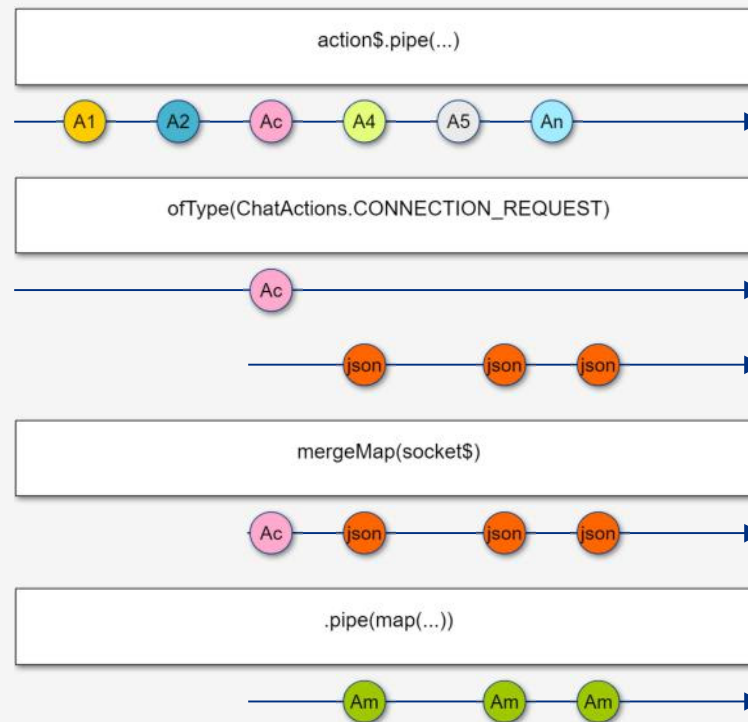
REDUX-OBSERVABLE IN PRACTICE

registrationRequestEpic (spaEpic.ts)



REDUX-OBSERVABLE IN PRACTICE

connectEpic (chatEpic.ts)



ROUTING



React Router



Aviator



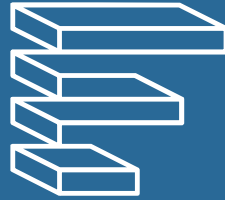
Backbone Router

i React's website has a more comprehensive list of available routers: [Link](#)

FORMS



Redux Forms

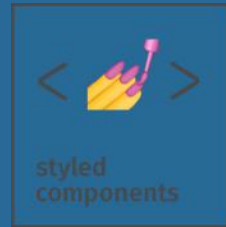


Formik

Formsy

i React's basic documentation on controlled and uncontrolled forms: [Link](#)

STYLING



Global Styles or
Imported Component Styles

React Specific Approaches



CSS Modules

TESTING



Jest



Enzyme

WHO IS REACT GOOD FOR?

- Existing projects
- Strong development teams which want flexibility
- Small projects

THANKS!

- Questions?