

EUGENE ZHARKOV

SWITCHING TO REACT.JS

FROM ANGULARJS DEVELOPER



Michael Jackson @mjackson · Jun 16

In my recent @reactjs code I find myself preferring createClass more and more over ES6 classes. So simple, no hacks required.



18



36



Ben Alpert

@soprano



Follow

@mjackson @reactjs Notice how we never told you to use ES6 classes? :)

RETWEETS

3

FAVORITES

10



10:52 AM - 16 Jun 2015



Michael Jackson @mjackson · Jun 16

@soprano yes, I did :) @reactjs



ES5

```
var Photo = React.createClass({  
  handleDoubleTap: function(e) { ... },  
  render: function() { ... },  
});
```

ES6+

```
class Photo extends React.Component {  
  handleDoubleTap(e) { ... }  
  render() { ... }  
}
```

COMPONENT INITIALIZATION

ES5

```
var EmbedModal = React.createClass({  
  componentWillMount: function() { ... },  
});
```

ES6+

```
class EmbedModal extends React.Component  
{  
  constructor(props) {  
    super(props);  
  
    // default state props  
  }  
}
```

ES5

```
var PostInfo = React.createClass({
  handleOptionsButtonClick: function(e) {
    this.setState({showOptionsModal: true});
  },
});
```

ES6+

```
class PostInfo extends React.Component {
  constructor(props) {
    super(props);

    this.handleOptionsButtonClick =
this.handleOptionsButtonClick.bind(this);
  }

  handleOptionsButtonClick(e) {
    this.setState({showOptionsModal: true});
  }
}
```

ES6+

```
class PostInfo extends React.Component {  
  handleOptionsButtonClick = (e) => {  
    this.setState({showOptionsModal: true});  
  }  
}
```

ES6+

```
class AutoloadingPostsGrid extends React.Component {  
  render() {  
    var {  
      className,  
      ...others, // contains all properties of this.props except for className  
    } = this.props;  
    return (  
      <div className={className}>  
        <PostsGrid {...others} />  
        <button onClick={this.handleClick}>Load more</button>  
      </div>  
    );  
  }  
}
```

BUILD TOOLS

- ▶ babel
- ▶ browserify
 - ▶ babelify (babel transpiler)
 - ▶ watchify (files watch)
 - ▶ factor-bundle (bundle splitting)
 - ▶ deAMDify (AMD support)
- ▶ webpack...

BROWSERIFY CLI EXAMPLE

```
watchify -t babelify app/js/index.js -o public/js/bundle.js
```

BROWSERIFY JS EXAMPLE

```
browserify({ debug: true })  
  .transform(babelify);
```

```
var fs = require("fs");
```

```
var browserify = require("browserify");
```

```
var babelify = require("babelify");
```

```
browserify({ debug: true })
```

```
  .transform(babelify)
```

```
  .require("./script.js", { entry: true })
```

```
  .bundle()
```

```
  .on("error", function (err) { console.log("Error: " + err.message); })
```

```
  .pipe(fs.createWriteStream("bundle.js"));
```

WEB PACK JS EXAMPLE

```
module: {  
  loaders: [  
    { test: /\.js$/, exclude: /node_modules/, loader: "babel-loader"}  
  ]  
}
```

BYE BYE DIRECTIVES & CONTROLLERS

TRUE LIE

ANGULAR DIRECTIVE BUTTHURT

```
myModule.directive('directiveName', function factory(injectables) {
  var directiveDefinitionObject = {
    priority: 0,
    template: '<div></div>', // or // function(tElement, tAttrs) { ... },
    // or templateUrl: 'directive.html', // or // function(tElement, tAttrs) { ... },
    transclude: false,
    restrict: 'A',
    templateNamespace: 'html',
    scope: false,
    controller: function($scope, $element, $attrs, $transclude, otherInjectables) { ... },
    controllerAs: 'stringIdentifier',
    bindToController: false,
    require: 'siblingDirectiveName', // or // ['^parentDirectiveName', '?optionalDirectiveName', '?
^optionalParent'],
    compile: function compile(tElement, tAttrs, transclude) {
      return {
        pre: function preLink(scope, iElement, iAttrs, controller) { ... },
        post: function postLink(scope, iElement, iAttrs, controller) { ... }
      }
      // or // return function postLink( ... ) { ... }
    },
    // or // link: {
    //   pre: function preLink(scope, iElement, iAttrs, controller) { ... },
    //   post: function postLink(scope, iElement, iAttrs, controller) { ... }
    // }
    // or // link: function postLink( ... ) { ... }
  };
  return directiveDefinitionObject;
});
```

ANGULAR DIRECTIVE BUTTHURT

`$compile`

`$scope`

`$element`

`$attrs`

`$transclude`

`$watch`

REACT.JS SOBRIETY

```
export default class Feature extends React.Component {  
  componentDidMount () { ... }  
  componentWillReceiveProps () { ... }  
  shouldComponentUpdate () { ... }  
  componentWillUpdate () { ... }  
  componentDidUpdate () { ... }  
  componentWillUnmount () { ... }  
  render() { ... }  
}
```

JSX

```
var Nav, Profile;
```

```
// Input (JSX):
```

```
var app = <Nav color="blue"><Profile>click</Profile></Nav>;
```

```
// Output (JS):
```

```
var app = React.createElement(  
  Nav,  
  {color:"blue"},  
  React.createElement(Profile, null, "click")  
);
```

REACT.JS COMPONENT EXAMPLE

```
export default class CaseStudyHeader extends React.Component {  
  render() {  
    let headerStyle = {color: this.props.globalStyles.color};  
    let containerStyle = {backgroundColor: this.props.bgColor};  
    return <div className="case-study-header" style={containerStyle}>  
      <h5 style={headerStyle}>Case Study</h5>  
      <h2>{this.props.caption}</h2>  
      <p dangerouslySetInnerHTML={{__html: this.props.bodyText}}></p>  
      <img src={this.props.imageSrc} />  
    </div>  
  }  
}
```

REACT.JS PARENT COMPONENT EXAMPLE

```
export default class HomePage extends React.Component {

  render() {
    return <article>
      { Data.page.map(function (item, i) {

        switch(item.type) {
          case 'caseStudyHeader':

            return <CaseStudyHeader {...item} globalStyles={Data.globalStyles} key={i} />

          case 'centeredImageBlock':
            return <CenteredImageBlock {...item} globalStyles={Data.globalStyles} key={i} />

          case 'notaBene':
            return <NotaBeneBlock {...item} globalStyles={Data.globalStyles} key={i} />

          case 'imageSeparator':
            return <ImageSeparator {...item} globalStyles={Data.globalStyles} key={i} />

          case 'article':
            return <Article {...item} globalStyles={Data.globalStyles} key={i} />
        }
      }, this)}
    </article>
  }
}
```

REACT.JS 'CUSTOM CONTROL' USAGE

```
<Radiogroup options={RADIOGROUP_YES_NO}  
  onChange={this.onSomeChange.bind(this)} />
```

REACT.JS 'CUSTOM CONTROL'

```
export default class Radiogroup extends React.Component {

  onChange(e) {
    this.props.onChange(e.currentTarget.value);
  }

  render () {

    let source = this.props.options;
    let name = shortid.generate();

    return <div>
      {source.map(function (item, i) {

        let id = name + i;

        return <span key={i}><input type="radio" name={name} id={id} value={item.value}
          onChange={this.onChange.bind(this)} />

          <label htmlFor={id}><span className="control"></span> {item.title}</label></span>

      }, this)}}

    </div>
  }
}
```

INLINE STYLES

```
var divStyle = {  
  color: 'white',  
  backgroundImage: 'url(' + imgUrl + ')',  
  WebkitTransition: 'all', // note the capital 'W' here  
  msTransition: 'all' // 'ms' is the only lowercase vendor prefix  
};
```

```
ReactDOM.render(<div style={divStyle}>Hello World!</div>, mountNode);
```

THEME MANAGER

```
checkbox: {
  boxColor: rawTheme.palette.textColor,
  checkedColor: rawTheme.palette.primary1Color,
  requiredColor: rawTheme.palette.primary1Color,
  disabledColor: rawTheme.palette.disabledColor,
  labelColor: rawTheme.palette.textColor,
  labelDisabledColor: rawTheme.palette.disabledColor,
},

datePicker: {
  color: rawTheme.palette.primary1Color,
  textColor: rawTheme.palette.alternateTextColor,
  calendarTextColor: rawTheme.palette.textColor,
  selectColor: rawTheme.palette.primary2Color,
  selectTextColor: rawTheme.palette.alternateTextColor,
}
```

COLORS

```
palette: {  
  
    primary1Color: Colors.cyan500,  
    primary2Color: Colors.cyan700,  
    primary3Color: Colors.lightBlack,  
    accent1Color: Colors.pinkA200,  
    accent2Color: Colors.grey100,  
    accent3Color: Colors.grey500,  
    textColor: Colors.darkBlack,  
    alternateTextColor: Colors.white,  
    canvasColor: Colors.white,  
    borderColor: Colors.grey300,  
    disabledColor: ColorManipulator.fade(Colors.darkBlack, 0.3),  
  
}
```

ICON-BUTTON

```
getStyles() {  
  
  const {  
    iconSize,  
    textColor,  
    disabledColor,  
  } = this.constructor.getRelevantContextKeys(this.state.muiTheme);  
  
  let styles = {  
    root: {  
      position: 'relative',  
      boxSizing: 'border-box',  
      transition: Transitions.easeOut(),  
      padding: iconSize / 2,  
      width: iconSize * 2,  
      height: iconSize * 2,  
      fontSize: 0,  
    }  
  }  
}
```


POSTCSS

- ▶ PostCSS itself is very small. It includes only a CSS parser, a CSS node tree API, a source map generator, and a node tree stringifier.
- ▶ All of the style transformations are performed by plugins, which are plain JS functions. Each plugin receives a CSS node tree, transforms it & then returns the modified tree.

POSTCSS CLI

```
postcss -o public/css/style.css -u precss -s postcss-scss app/css/index.scss -w
```

ANGULARJS ROUTING

```
myApp.config(function($stateProvider, $urlRouterProvider) {

  $urlRouterProvider.otherwise("/state1");
  $stateProvider
    .state('state1', {
      url: "/state1",
      templateUrl: "partials/state1.html"
    })

    .state('state1.list', {
      url: "/list",
      templateUrl: "partials/state1.list.html",
      controller: function($scope) {
        $scope.items = ["A", "List", "Of", "Items"];
      }
    })

    .state('route2', {
      url: "/route2",
      views: {
        "viewA": { template: "route2.viewA" },
        "viewB": { template: "route2.viewB" }
      }
    })
});
```

REACT.JS ROUTING

```
<Router>
```

```
  <Route path="/" component={App}>
```

```
    <Route path="about" component={About}/>
```

```
    <Route path="users" component={Users}>
```

```
      <Route path="/user/:userId" component={User}/>
```

```
    </Route>
```

```
    <Route path="*" component={NoMatch}/>
```

```
  </Route>
```

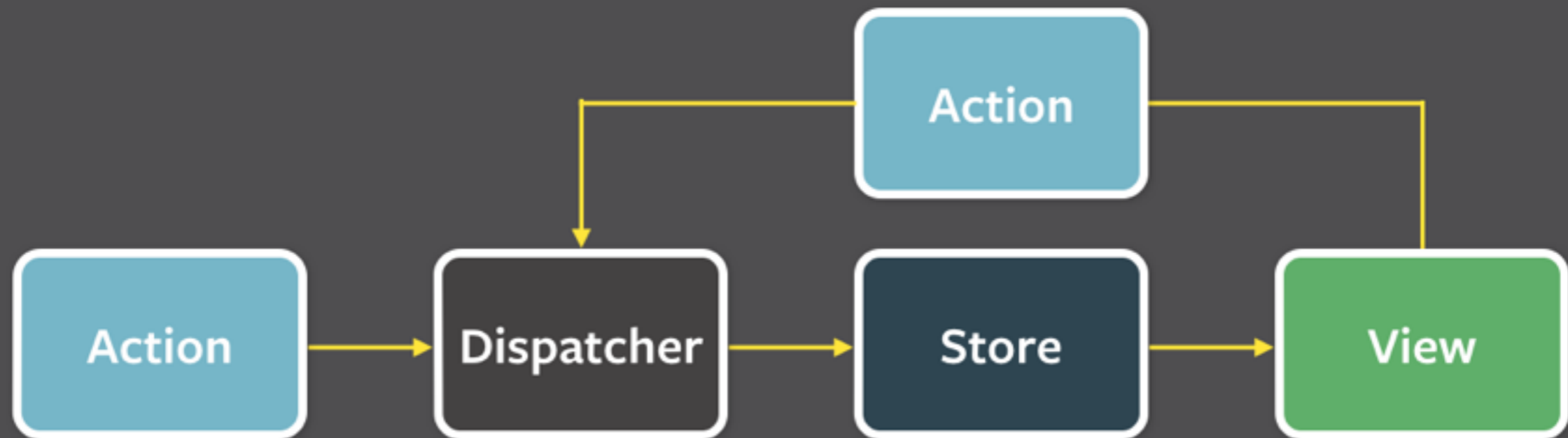
```
</Router>
```

REACT.JS ROUTING HISTORY

```
const createBrowserHistory = require('history/lib/createBrowserHistory');

ReactDOM.render ((
  <Router history={createBrowserHistory()}>
    ...
  </Router>
), document.body);
```

FLUX



FLUX

- ▶ Single Dispatcher
 - ▶ Central hub that manages all data flow. A Simple mechanism for distributing the actions to the stores.
- ▶ Stores
 - ▶ Stores contain the application state and logic. Their role is somewhat similar to a model in a traditional MVC, but they manage the state of many objects – they do not represent a single record of data like ORM models do.

FLUX

▶ Actions

- ▶ The dispatcher exposes a method that allows us to trigger a dispatch to the stores, and to include a payload of data, which we call an action.

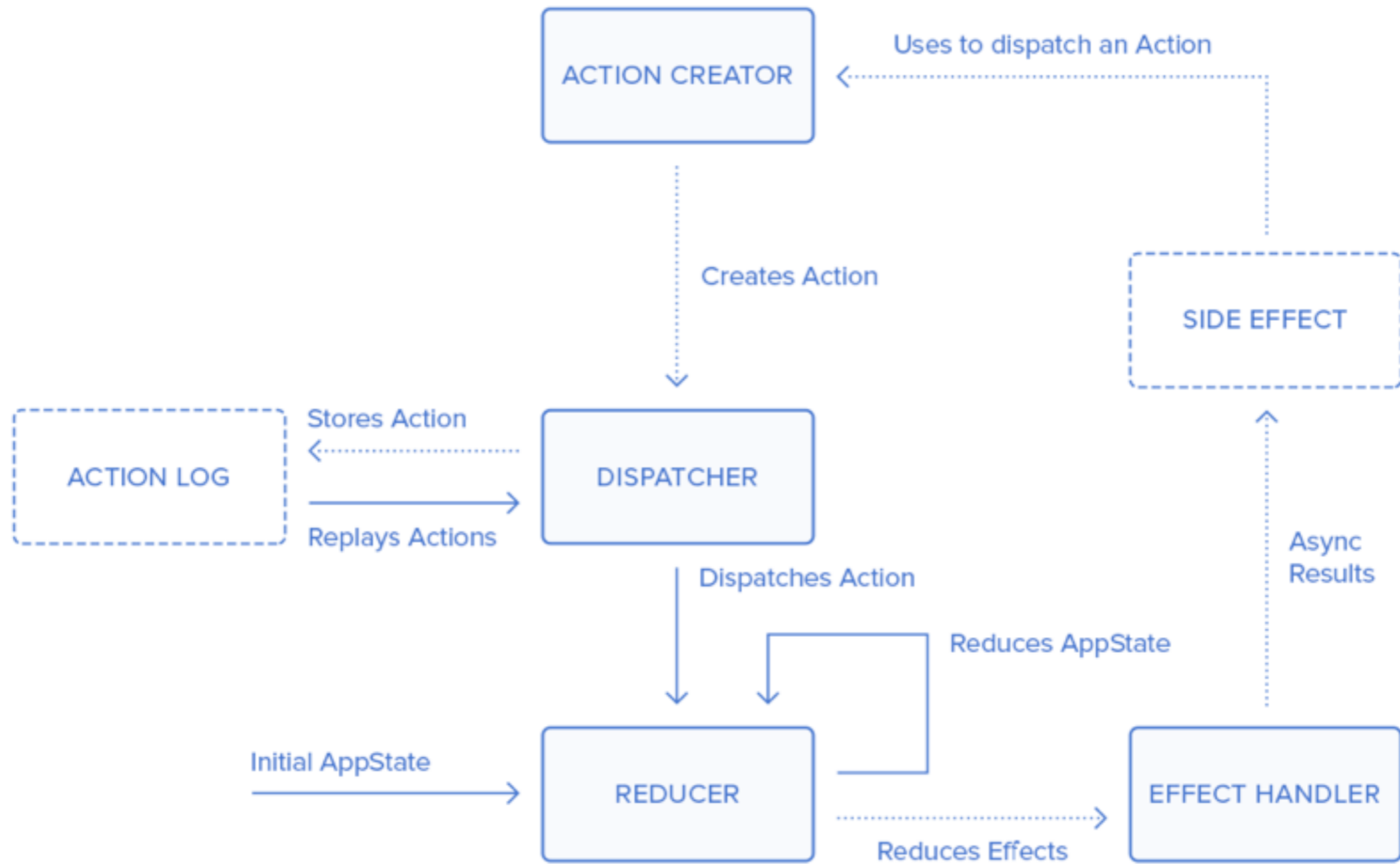
▶ Views

- ▶ When it receives the event from the store, it first requests the new data it needs via the stores' public getter methods. It then calls its own `setState()` or `forceUpdate()` methods, causing its `render()` method and the `render()` method of all its descendants to run.

FLUX

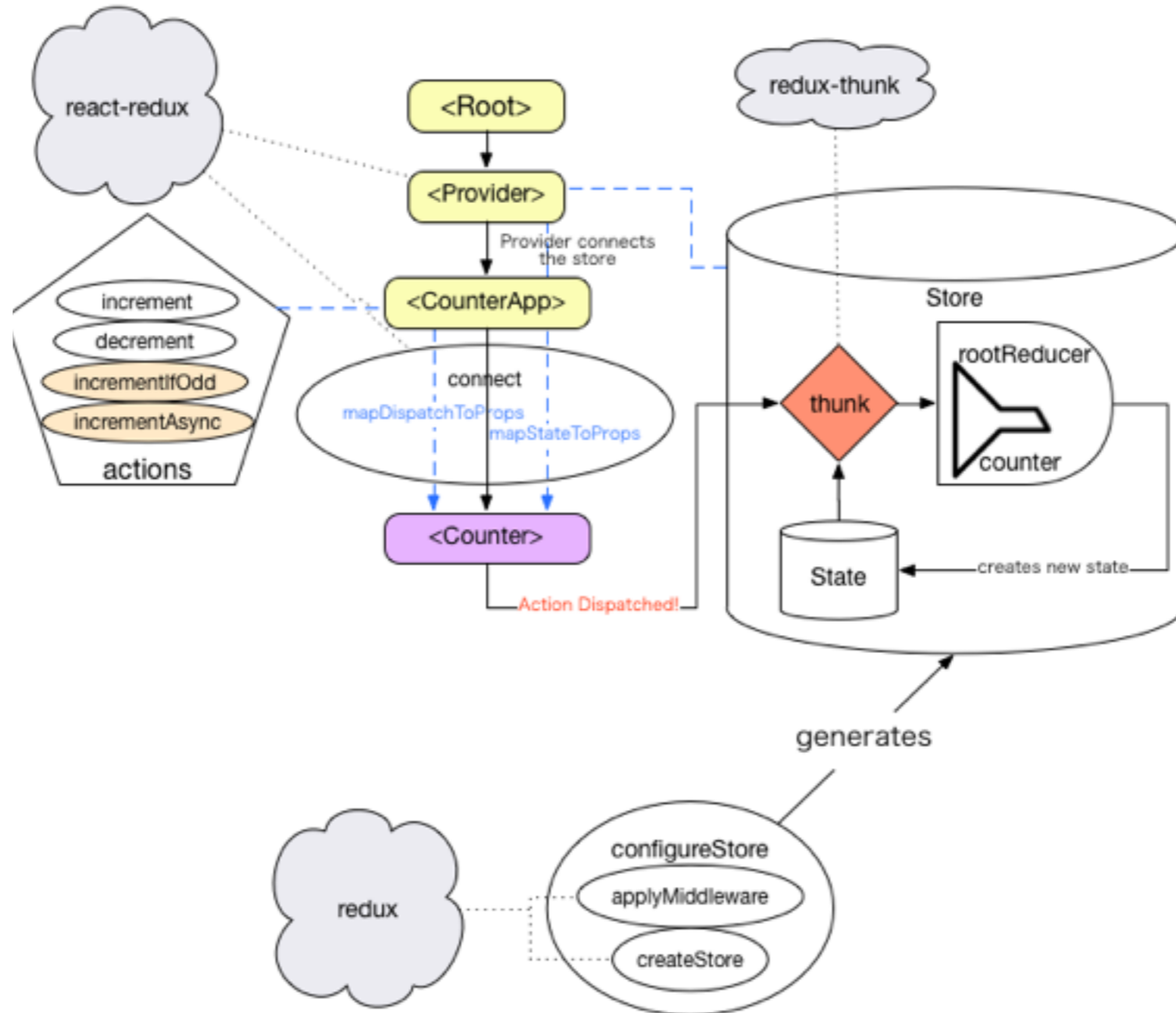
- ▶ myapp
- ▶ ...
 - ▶ js
 - ▶ actions
 - ▶ components
 - ▶ constants
 - ▶ dispatcher
 - ▶ stores
- ▶ index.html

REDUX



→ REPLAYABLE → IGNORING REPLAY

REDUX, MORE SCARY DIAGRAM



REDUX PRINCIPLES

- ▶ Single store = Single application state
- ▶ Read-only state
- ▶ Mutations are written as pure functions

REDUX

- ▶ myapp
 - ▶ js
 - ▶ actions
 - ▶ components
 - ▶ constants
 - ▶ reducers
 - ▶ routes
 - ▶ stores
- ▶ index.html

REDUCER

```
function posts(state = {
  isFetching: false, didInvalidate: false, items: [] }, action) {

  switch (action.type) {

    case INVALIDATE_REDDIT:
      return Object.assign({}, state, {
        didInvalidate: true
      });

    case REQUEST_POSTS:
      return Object.assign({}, state, {
        isFetching: true,
        didInvalidate: false
      });
    case RECEIVE_POSTS:
      return Object.assign({}, state, {
        isFetching: false,
        didInvalidate: false,
        items: action.posts,
        lastUpdatedAt: action.receivedAt
      });

    default:
      return state;
  }
}
```

ACTIONS

```
function requestPosts(reddit) {  
  return {  
    type: REQUEST_POSTS,  
    reddit  
  };  
}
```

```
function receivePosts(reddit, json) {  
  
  return {  
    type: RECEIVE_POSTS,  
    reddit,  
    posts: json.data.children.map(child => child.data),  
    receivedAt: Date.now()  
  };  
}
```

```
export function fetchPosts(reddit) {  
  
  return dispatch => {  
    dispatch(requestPosts(reddit));  
    return fetch(`http://www.reddit.com/r/${reddit}.json`)  
      .then(req => req.json())  
      .then(json => dispatch(receivePosts(reddit, json)));  
  };  
}
```

STORE

```
import { createStore, applyMiddleware } from 'redux';
import thunkMiddleware from 'redux-thunk';
import createLogger from 'redux-logger';
import rootReducer from './reducers';

const loggerMiddleware = createLogger();
const createStoreWithMiddleware = applyMiddleware(

  thunkMiddleware,
  loggerMiddleware

)(createStore);

export default function configureStore(initialState) {

  return createStoreWithMiddleware(rootReducer, initialState);

}
```


ROOT OBJECT

```
import React, { Component } from 'react';
import { Provider } from 'react-redux';
import configureStore from '../configureStore';
import AsyncApp from './AsyncApp';
```

```
const store = configureStore();
```

```
export default class Root extends Component {
  render() {
    return (
      <Provider store={store}>
        <AsyncApp />
      </Provider>
    );
  }
}
```

SMART COMPONENT

```
import React, { Component, PropTypes } from 'react';
import { connect } from 'react-redux';
import { selectReddit, fetchPosts, invalidateReddit } from '../actions';

class AsyncApp extends Component {
  constructor(props) {
    super(props);
    this.handleChange = this.handleChange.bind(this);
    this.handleRefreshClick = this.handleRefreshClick.bind(this);
  }

  componentDidMount() {
    const { dispatch, selectedReddit } = this.props;
    dispatch(fetchPosts());
  }

  handleChange(nextReddit) {
    this.props.dispatch(selectReddit(nextReddit));
  }

  render () {
    const { selectedReddit, posts, isFetching, lastUpdated } = this.props;
    return (.....)
  }
}
```

SMART COMPONENT

```
function mapStateToProps(state) {
  const { selectedReddit, postsByReddit } = state;
  const {
    isFetching,
    lastUpdated,
    items: posts
  } = postsByReddit[selectedReddit] || {
    isFetching: true,
    items: []
  };

  return {
    selectedReddit,
    posts,
    isFetching,
    lastUpdated
  };
}

export default connect(mapStateToProps)(AsyncApp);
```

**REDUX? I KNOW NOTHING
ABOUT REDUX.**

DUMB COMPONENT

REDUX THUNK

- ▶ Redux Thunk middleware allows you to write action creators that return a function instead of an action.

LINKS

- ▶ [davezuko / react-redux-starter-kit](#)
- ▶ [emmenko / redux-react-router-async-example](#)
- ▶ [official documentation](#)

LINKS

- ▶ [davezuko / react-redux-starter-kit](#)
- ▶ [emmenko / redux-react-router-async-example](#)
- ▶ [official documentation](#)
- ▶ [slides](#)



QUESTIONS



THANK YOU

@2J2E

EU.ZHARKOV@GMAIL.COM