

Building Applications with React and Flux



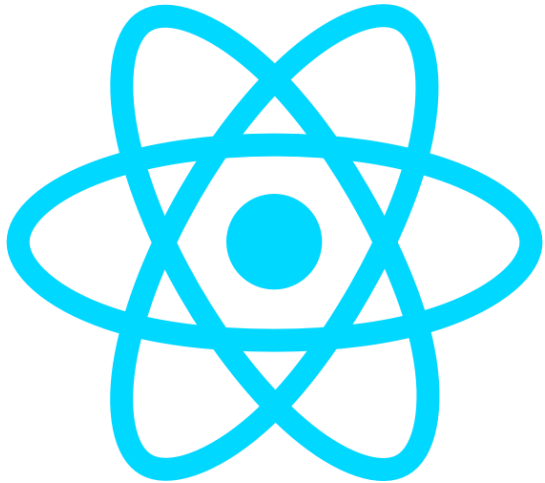
Cory House

PRINCIPAL CONSULTANT

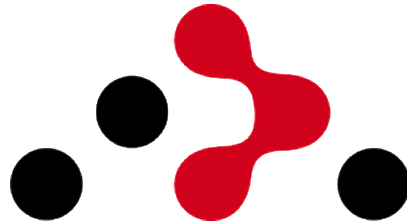
reactjsconsulting.com



Who is This Course For?



New to React



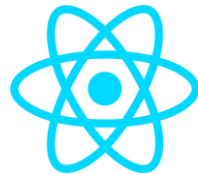
New to React Router



New to Flux



Here's the Plan



Intro and Core Technologies

Environment Setup

React Core Concepts

Creating Components

Props, State, Lifecycle and Keys

Hooks, Component Composition, PropTypes

React Router 

React Forms



Flux

Flux Demos



Agenda



Who is this course for?

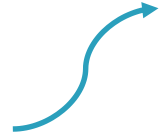
Why React?

Tech overview



Building Applications with React and Flux

Intro



Introduces React
Flux

create-react-app

Introduces React Router

No Testing

Course Admin App

Building Applications with React and Redux

Assumes you know React



Advanced

Redux

Custom dev environment

Assumes you know React Router

Jest, Enzyme, React Testing Library

Course Admin App



What about testing?



Automated testing

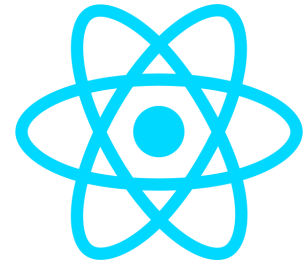
Courses with testing coverage

- Building Applications with React and Redux
- Creating Reusable React Components

Why React?



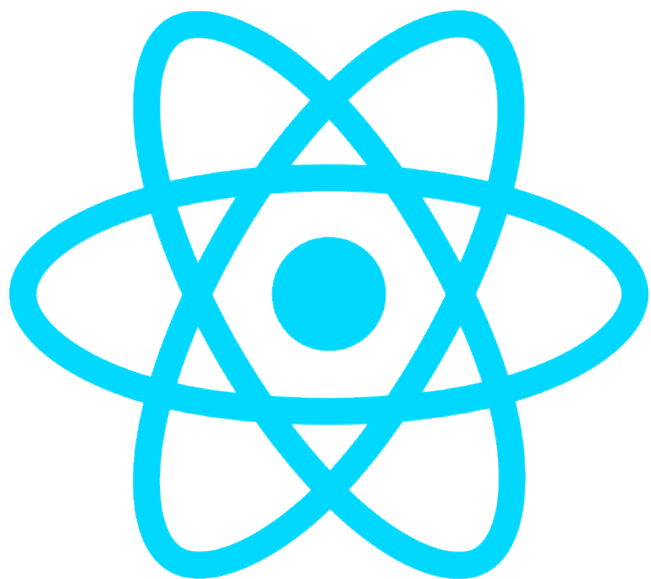
A World of Components



Kendo UI
THE ART OF WEB DEVELOPMENT



React



Component library
Simple composition



Why React?



Fast

Composable

Pluggable

Isomorphic friendly

Simple

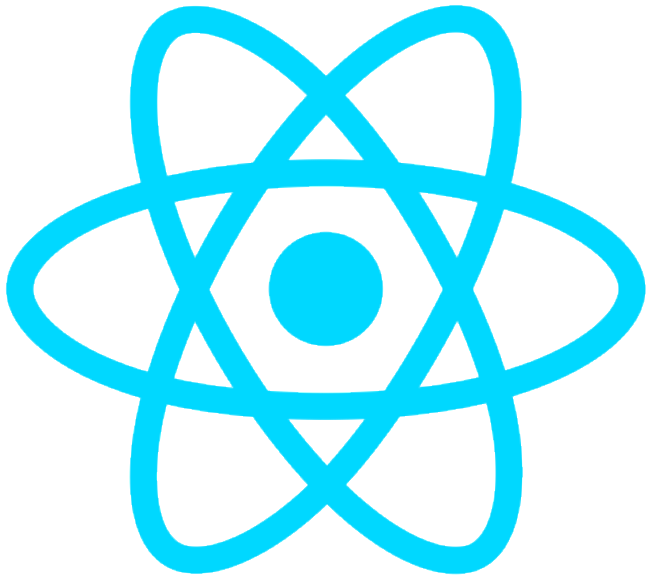
Battle proven



Companies Using React



Innovation



JSX

Virtual DOM

Isomorphic Rendering

Unidirectional Flows



They call it **React** for a reason.





Ben Alman

@cowboy



Follow

Facebook: Rethink established best practices™



RETWEETS

21

FAVORITES

21



4:40 PM - 29 May 2013





Try to keep an **open** mind



HTML should be a *projection* of app state,
not a source of truth.



JavaScript and HTML belong in the *same file*.



Unidirectional flow. No two-way binding.



Inline styles can be *good*.



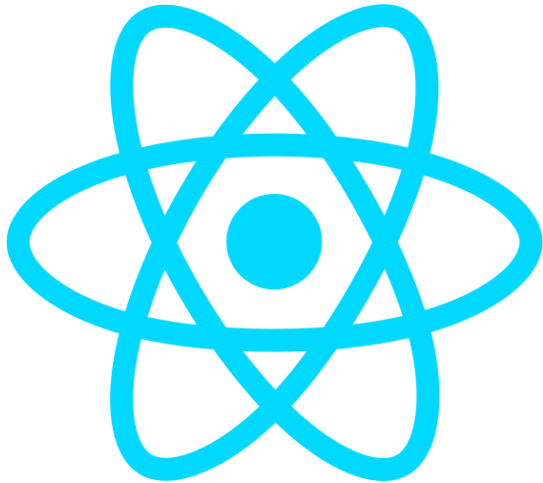
~~All variables should be global.~~



Tech Overview



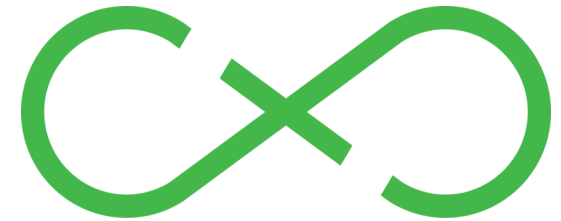
Core Technologies



React
Components



React Router
Routing



Flux
Data flow



React Router



Client-side router

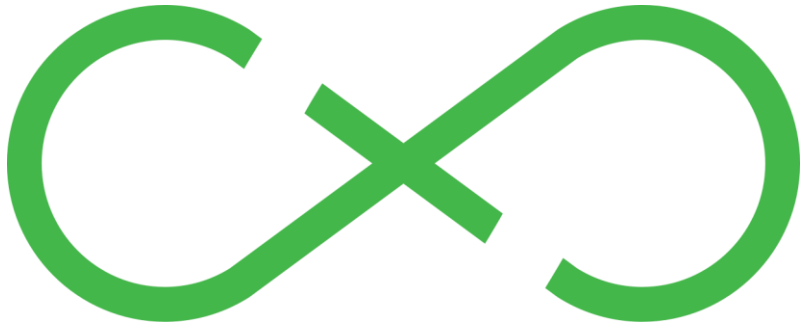
Instantaneous page transitions

Declarative

Nested views map to nested routes



Flux



Uni-directional data flows

Updates flow in one direction

Easy to debug and scale

More a pattern than a library

Redux is a popular “flavor”





React Starter Projects

These are repositories that you copy and modify to create your own React app. Pick a starter project with all the features you need, and none that you don't.

To get started with most of these, you would clone the repository and then start making modifications.

Other starters are installed rather than copied: they become a dependency of your project. These [dependency starters](#) make it easier to get updates and they keep your project directory cleaner (because much of the code is wrapped in the dependency).

TAGS

AGE

198 Projects

Sort ▼



create react app



Option: Build a dev environment.

Why?

1. Understand how key technologies work together
2. Learn how to customize your dev environment





What do you want to learn?



Building Applications with React and Redux

by Cory House

Learn how to use Redux, React Router, and ES6 to build a real world app with React. Use Webpack, Babel, ESLint, npm scripts, Mocha, Enzyme, and more to build a rich, one step, custom React development environment and build process from the ground up.

 Resume Course



Bookmark



Add to Channel



Download Course



Table of contents

Description

Transcript

Exercise files

Discussion

Learning Check

Recommended

These exercise files are intended to provide you with the assets you need to create a video-based hands-on experience. With the exercise files, you can follow along with the author and re-create the same solution on your computer. We find this to be even more effective than written lab exercises.

Download exercise files

Course author



Cory House

Cory is an independent consultant with over 15 years of experience in software development. He is the principal consultant at reactjsconsulting.com and a Microsoft MVP.

Course info

Level Intermediate

Rating ★★★★★ (1183)

My rating ★★★★★

Share course





Create React App

Set up a modern web app by running one command.



GET STARTED

Less to Learn

You don't need to learn and configure many build tools. Instant reloads help you focus on development. When it's time to deploy, your bundles are optimized automatically.

Only One Dependency

Your app only needs one build dependency. We test Create React App to make sure that all of its underlying pieces work together seamlessly – no complicated version mismatches.

No Lock-In

Under the hood, we use Webpack, Babel, ESLint, and other amazing projects to power your app. If you ever want an advanced configuration, you can “eject” from Create React App and edit their config files directly.

Dev Environment: Create React App



Nodel



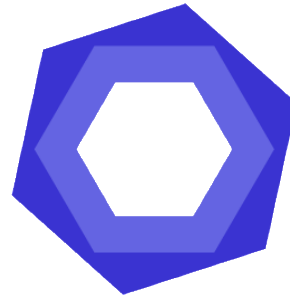
npm



Babel



Webpack



ESLint



Jest



Node



Server-side JS

Uses the V8 Engine

Includes npm package manager



npm: Node's Package Manager



.NET



Ruby



Java

maven

Python

PIP

PHP



Node



Babel



Transpile our code

We can use the latest JS features



ECMAScript Versions

1997 **ES1**

1998 **ES2**

1999 **ES3**

Tons of features! Yearly releases begin!

2009 **ES5**

2015 **ES2015**

Destructuring, arrows, const, let, classes...

2016 **ES2016**

Exponent operator, array.includes.

2017 **ES2017**

Async/Await

2018 **ES2018**

Rest/Spread, dynamic imports

2019 **ES2019**



Sort by Engine types

Show obsolete platforms

Show unstable platforms

V8

SpiderMonkey

JavaScriptCore

Chakra

Carakan

KJS

Other

Minor difference (1 point)

Small feature (2 points)

Medium feature (4 points)

Large feature (8 points)

		Compilers/polyfills							Desktop browsers											
		98%	56%	70%	70%	50%	68%	17%	5%	11%	96%	96%	96%	98%	98%	98%	98%	98%	98%	98%
Feature name	Current browser	Traceur	Babel 6+ core-js	Babel 7+ core-js	Closure 2018.11	TypeScript + core-js	es6-shim	Konq 4.14 ^[1]	IE 11	Edge 17	Edge 18	Edge 19 Preview	FF 60 ESR	FF 63	FF 64	FF 65 Beta	FF 66 Nightly	CH 70 OP 57	CH 71 OP 58	CH 72 OP 59
Optimisation																				
• proper tail calls (tail call optimisation)	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2
Syntax																				
• default function parameters	7/7	4/7	4/7	4/7	5/7	5/7	0/7	0/7	0/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7
• rest parameters	5/5	4/5	3/5	3/5	2/5	4/5	0/5	0/5	0/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5
• spread syntax for iterable objects	15/15	15/15	13/15	13/15	11/15	14/15	0/15	0/15	0/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15
• object literal extensions	6/6	6/6	6/6	6/6	5/6	6/6	0/6	0/6	0/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6
• for..of loops	9/9	9/9	9/9	9/9	6/9	9/9	0/9	0/9	0/9	9/9	9/9	9/9	9/9	9/9	9/9	9/9	9/9	9/9	9/9	9/9
• octal and binary literals	4/4	2/4	4/4	4/4	2/4	4/4	2/4	0/4	0/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4
• template literals	5/5	4/5	4/5	4/5	3/5	3/5	0/5	0/5	0/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5
• RegExp "y" and "u" flags	6/6	4/6	4/6	4/6	0/6	0/6	0/6	0/6	0/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6
• destructuring, declarations	22/22	20/22	21/22	21/22	20/22	21/22	0/22	0/22	0/22	22/22	22/22	22/22	22/22	22/22	22/22	22/22	22/22	22/22	22/22	22/22
• destructuring, assignment	24/24	23/24	24/24	24/24	22/24	24/24	0/24	0/24	0/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24
• destructuring, parameters	24/24	19/24	21/24	21/24	20/24	21/24	0/24	0/24	0/24	23/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24
• Unicode code point escapes	2/2	1/2	1/2	1/2	1/2	1/2	0/2	0/2	0/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2
• new.target	2/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2
Bindings																				
• const	18/18	16/18	14/18	14/18	16/18	14/18	0/18	2/18	14/18	18/18	18/18	18/18	18/18	18/18	18/18	18/18	18/18	18/18	18/18	18/18
• let	14/14	12/14	10/14	10/14	12/14	10/14	0/14	0/14	12/14	14/14	14/14	14/14	14/14	14/14	14/14	14/14	14/14	14/14	14/14	14/14
• block-level function declaration ^[17]	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Webpack



Bundler

Use npm packages on web apps

Bundle styles, images and more



ESLint



Check our code for issues

Report issues on the command line

Highly configurable and extendable



Wrap Up



Why React?

- Fast, composable, simple, proven
- JSX, virtual DOM

Core Tech

- React – Components
- React Router – Routing
- Flux – Data flows
- Node, npm – Packages/automation
- Babel – Transpile
- Webpack - Bundle
- ESLint – Lint

Next up: Set up dev environment

