

---

## React/Redux Links

Curated tutorial and resource links I've collected on React, Redux, ES6, and more, meant to be a collection of high-quality articles and resources for someone who wants to learn about the React-Redux ecosystem, as well as a source for quality information on advanced topics and techniques. Not quite "awesome", but hopefully useful as a starting point I can give to others. Suggestions welcome.

Another important resource is the **Reactiflux community on Discord**, which has chat channels dedicated to discussion of React, Redux, and other related technologies. There's always a number of people hanging out and answering questions, and it's a great place to ask questions and learn. The invite link is at **<https://www.reactiflux.com>**.

You might also want to check out my categorized list of Redux-related addons, libraries, and utilities, at Redux Ecosystem Links. Also see Community Resources for links to other links lists, podcasts, and email newsletters. Finally, I also keep a dev blog at [blog.isquaredsoftware.com](http://blog.isquaredsoftware.com), where I write about React, Redux, Webpack, and more.

## Table of Contents

### Getting Started

- Basic Concepts and Overviews
- Community Resources
- Javascript Resources
- Git Resources and Tutorials
- Node.js and NPM
- Webpack Tutorials
- Boilerplates and Starter Kits

### Basic Tutorials

- ES6 Features and Samples
- React Tutorials
- Flux Tutorials
- Redux Tutorials
- MobX Tutorials

### Intermediate Concepts

- Using React with ES6

- 
- Functional Programming
  - Immutable Data
  - React/Redux Testing
  - React Native
  - React Tips and Techniques

## **Advanced Topics**

- **Architecture and Structure**

- Project Structure
- React Component Patterns
- React Component Composition
- React State Management
- React Architecture and Best Practices
- Redux Architecture and Best Practices
- React/Redux Performance
- React Deployment

- **React:**

- React Implementation and Concepts
- React and Forms
- React and AJAX
- React Styling
- React Server Rendering
- React and Routing

- **Redux**

- Redux Reducers and Selectors
- Redux Middleware
- Redux Side Effects
- Redux UI Management
- Redux Tips and Techniques
- Using Redux Without React

- **Other**

- Webpack Advanced Techniques
- Static Typing

---

## Comparisons and Discussion

- [React/Flux/Redux Pros, Cons, and Discussion](#)
- [Framework Comparisons](#)

## Recommended Learning Path

**You should usually learn these technologies in the following order:**

1. **“How Web Apps Work”**: a series of posts that lays out the big picture of the core technologies, terms, and concepts used in client/server web apps
2. **JavaScript**: If you don’t know JavaScript, nothing else will make sense
3. **React**: You can use React by itself, or with Redux and/or TypeScript. Learning it separately will minimize the number of new concepts and syntax you have to learn at once.
4. **Redux**: Redux can be used separately, but it’s most commonly used with React.
5. **TypeScript**: Because it adds static types on top of JS, you need to understand JS first. Also, it’s easiest to understand React and Redux first, *then* learn how to use them with static types.

The resources in this page are listed in that order.

**You are not required to read every single link and article listed in this page.** However, you should try to read through as many of the articles linked in the “Recommended Primary Resources” sections as possible, especially for topics you are not already familiar with. Many of the recommended tutorials do cover the same topics, so feel free to skip past concepts you’ve already learned.

Links in the “Additional Resources” sections are available as references and reading as needed.

## How Web Apps Work

Mark’s post series that describes the key terms, concepts, technologies, syntax, and data flow used in web apps.

## Recommended Primary Resources (should read)

- [How Web Apps Work: HTTP and Servers](#)
- [How Web Apps Work: Client Development and Deployment](#)
- [How Web Apps Work: Browsers, HTML, and CSS](#)
- [How Web Apps Work: JavaScript and the DOM](#)
- [How Web Apps Work: AJAX, APIs, and Data Transfer](#)

---

## Javascript

### Recommended Primary Resources (should read)

#### General JS

- **Slides:** Mark's "JavaScript for Java Developers" slides
- **Read:** MDN: A re-introduction to JavaScript
- **Read:** The Modern JavaScript Tutorial
- **Read:** Javascript Cheatsheet
- **Exercises:** CodeCademy - Introduction to JavaScript Tutorial

#### Specific Topics

- Array methods:
  - Modern JS Tutorial - Array Methods
  - Which Array Function When?
- Equality and Comparisons
  - MDN - Equality comparisons and sameness
  - JS Equality Comparison Table
- Closures
  - MDN - Closures
  - Modern JS Tutorial - Closures
- **this** keyword and scopes
  - A gentle explanation of **this** keyword in JavaScript
  - **this** in JavaScript
  - Everything you wanted to know about JavaScript scope
- The JS event loop
  - Watch: What the heck is the event loop anyway?
  - The JavaScript Event Loop
- CSS and layout
  - CSS: From Zero to Hero
  - MDN - Visual Formatting Model
  - MDN - Introduction to the CSS Box Model

- 
- HTML and CSS is Hard (but it doesn't have to be)
    - A Complete Guide to Flexbox
    - A Complete Guide to Grid
  - Node / NPM
    - How to install Node.js
    - An introduction to the npm package manager
  - Build Tools
    - The Many Jobs of JS Build Tools
  - Debugging
    - The definitive guide to debugging JavaScript
    - Intro to debugging React Applications

## **Additional Resources (read as needed)**

### **General JS**

- **Core References:**
  - MDN: JavaScript Reference
  - The Modern JavaScript Tutorial
  - Online book: JavaScript for Impatient Programmers
- **Syntax Overviews:**
  - Javascript Cheatsheet
  - ES6 Overview in 350 Bullet Points
  - ES6 Feature Examples
- **Additional Books / References:**
  - The Complete JavaScript Handbook
  - Eloquent JavaScript
  - Exploring JS books (cover what's new in each yearly revision of the language)
  - Links: ES6+ Features and Syntax (links to additional articles on new features in ES6+)
  - You Don't Know JS (advanced concepts and understanding of JS behavior)

---

## Specific Topics

- Array/object methods / immutability
  - Does It Mutate?
  - Array Explorer and Object Explorer
- JS Event Loop
  - The JavaScript Event Loop
- Regular Expressions
  - A Guide to JavaScript Regular Expressions
  - A Beginner's Guide to Regular Expressions in JavaScript
- CSS
  - Online Interactive CSS Cheat Sheet
  - A Practical CSS Cheat Sheet
  - GRID: A simple visual cheatsheet for CSS Grid Layout
- Node / NPM
  - Introduction to Node.js
  - The package.json guide
- Lodash
  - Lodash documentation
- Build Tools
  - Babel
    - \* Babel Docs
    - \* Babel Tutorial (TutorialsPoint)
    - \* A Beginner's Guide to Babel
    - \* A Short and Simple Guide to Babel
  - Webpack
    - \* Webpack docs
    - \* Webpack Academy
    - \* Webpack from First Principles

## React

### Recommended Primary Resources (should read)

---

**General React** Start with reading the official docs first. The React team is in the process of starting a major rewrite of the React docs site to focus on teaching function components and hooks first, which is now available in beta. We've linked to that rather than the existing "main" documentation.

These other listed tutorials are also excellent and may explain things in a different way.

- **Read: official React docs**
  - Getting Started (docs overview and related resources)
  - Main Concepts (read the whole series, but especially these two):
    - ★ Lifting State Up
    - ★ Thinking In React
  - React Hooks guide (lays out the motivation, teaches hooks, API reference, in-depth FAQ)
- **Read: React docs (converted to show hooks)**
  - Tutorial
- **Watch:** React Tutorial for Beginners
- **Read:** Intro to React, Redux, and TypeScript for 2020 (Mark's presentation slides)
- **Read:** Build a CRUD App in React with Hooks
- **Read:** A Comprehensive Guide to React in 2020
- **Exercises:** Learn React - Interactive Tutorials

## Project Setup

- **Read: Simple React Development in 2019** (a guide to setting up an app, development environment, and deployment)
- **Use: CodeSandbox.io** (an online IDE that uses VS Code's editor, and can let you develop and run your apps completely in the browser)
- **Use: Create-React-App** (the official CLI tool for creating a React app with one command. Sets up a project with good default build settings out of the box.)

## Specific Topics

- Understanding how React works conceptually / internally
  - React as a UI Runtime (deep dive - not *required* reading, but will definitely help you understand React better)
  - Mark Erikson: A (Mostly) Complete Guide to React Rendering Behavior

- 
- Build your own React
  - State and props
    - A Visual Guide to State in React
  - Component lifecycles
    - React component lifecycle interactive diagram
  - AJAX requests
    - AJAX Request in React - How and Where to Fetch Data
  - Immutability
    - Immutability in React and Redux: The Complete Guide
    - Redux docs: Immutable Update Patterns
  - Functional Programming basics
    - The Little Idea of Functional Programming
    - What is Functional Programming?
  - Forms and “controlled inputs”
    - React docs - Forms
    - Controlled and uncontrolled form inputs in React don’t have to be complicated
    - Transitioning from uncontrolled inputs to controlled
  - React’s new “hooks” API
    - React docs - Hooks (lays out the motivation, teaches hooks, API reference, in-depth FAQ)
    - A Complete Guide to useEffect (very long article, but a must-read. Teaches how hooks use closures, defining when effects run, and much more.)
    - What are React Hooks?
    - React Hooks: Not Magic, Just Arrays (looks under the hood to explain how hooks are implemented)

## **Additional Resources (read as needed)**

### **General React**

- **Resource Collections**
  - Mark Erikson’s React-Redux links collection (many categories of links to articles)

- 
- Mark's suggested resources for learning React
  - Dave Ceddia's blog (everything he's written)
  - Robin Wieruch's blog (also everything he's written)

- **Additional Books / References**

- The Road to React
  - Learn Pure React

## Redux

### Recommended Primary Resources (should read)

**General Redux** Start with reading the official docs first.

The other tutorials are also excellent and may explain things in a different way.

- **Read: Redux core docs**

- **"Redux Essentials" tutorial:** explains "how to use Redux, the right way", using the latest recommended techniques and practices like Redux Toolkit and the React-Redux API, while building a real-world-ish example app.
  - **"Redux Fundamentals" tutorial:** teaches "how Redux works, from the ground up". including core Redux data flow and why standard Redux patterns exist.
  - "Typescript quick start": explains how to configure Redux Toolkit with type safety from action creators through to selectors.

- **Use: Redux Toolkit** (an official Redux package to simplify common tasks, including store setup and writing reducers)

- Example project: <https://github.com/reduxjs/redux-essentials-example-app/tree/tutorial-steps>

- **Read: React-Redux docs**

- **React-Redux hooks API reference**
  - These APIs are now considered outdated, but are widely used in existing Redux codebases
    - \* `connect()`: Extracting Data with `mapStateToProps`
    - \* `connect()`: Dispatching Actions with `mapDispatchToProps`

- **Watch:** Dan Abramov's tutorial videos on Egghead

- Getting Started with Redux

- 
- Building React Apps with Idiomatic Redux
  - **Read:** A Complete React-Redux Tutorial
  - **Read:** React Redux Tutorial for Beginners: The Definitive Guide
  - **Read:** Leveling Up with React: Redux

### Mark Erikson's Redux Resources

- **Read: “Idiomatic Redux” concepts and opinion series.** A series of blog posts that describes standard Redux development best practices, why they exist, and how Redux is meant to be used. (These are not required reading to get started, but highly recommended once you understand the basics.)
- Legacy resources (do not cover “Modern Redux”, but still informative)
  - **Read: Redux Fundamentals Workshop slices:** a 2-day internal workshop that covers Redux from the ground up. Includes complete recordings of each section, slides, and an exercises repo. (Does not cover “Modern Redux”, but
  - **Read: “Practical Redux” blog tutorial series.** Covers multiple React and Redux concepts through building a larger example application

### Specific Topics

- Tradeoffs of using Redux:
  - Mark Erikson: When (and when not) to Reach for Redux
  - Dan Abramov: the Case for Flux
  - Dan Abramov: You Might Not Need Redux
- Reducer functions
  - Redux docs - Structuring Reducers
- Selector functions
  - Idiomatic Redux: Using Reselect Selectors for Encapsulation and Performance
- Side effects
  - Stack Overflow: How do we dispatch an action with a timeout?
  - Stack Overflow: Why do we need middleware for async flow?
  - What the heck is a “thunk”?
  - What is the right way to do asynchronous operations in Redux? (side-by-side comparison of multiple side effects approaches)

- 
- Redux Power Tools: Redux-Saga
  - Normalizing data
    - Redux docs - Normalizing State Shape
    - Normalizing Redux Stores for Maximum Code Reuse

### **Additional Resources (read as needed)**

- **Resource Collections**

- **Redux Style Guide:** best practices and recommendations for using Redux the right way
- **Redux FAQ** (answers to many common questions about Redux)
- Redux Ecosystem Links (a curated list of Redux-related addons and utilities)

- **Books and Courses**

- Pure Redux course by Dave Ceddia
- Redux course by Tyler McGinnis
- Learn Redux course by Wes Bos (free)
- Redux in Action
- The Complete Redux Book (free)
- Taming the State in React

### **Specific Topics**

- How does Redux work?
  - Build Yourself a Redux
  - Idiomatic Redux: The History and Implementation of React-Redux

## **TypeScript**

### **Recommended Primary Resources (should read)**

- **Read: official TypeScript docs**
- Typescript Playground - an interactive playground for testing typescript behavior and reproducing issues - includes some built-in examples
- **Read: Get Started with TypeScript in 2019**
- **Read: The Definitive TypeScript Guide**
  - TypeScript Cheat Sheet
- **Read: The TypeScript Guide**

---

## Specific Topics

- **interface** vs **type**
  - TypeScript Interface vs Type
    - ★ Interface vs Type sandbox example
- “Typescript quick start”: explains how to configure Redux Toolkit with type safety from action creators through to selectors.

## Additional Resources (read as needed)

- **Resource Collections**
  - **React+TypeScript Cheatsheets** (a definitive set of information on how to use TypeScript with React)
  - React + Redux in TypeScript - Static Typing Guide (a comprehensive set of info on using React, Redux, and TS together, with a focus on getting complete / “correct” type coverage of an app)
- **Techniques**
  - Redux with Code-Splitting and Type Checking