

# Managing State with Third-party Libraries



**Cory House**

React Consultant and Trainer

@housecor | reactjsconsulting.com



# Third Party State Libraries

**Redux**

**Mobx**

**Recoil**

**react-query**

**swr**

**Relay**

**Apollo**

**Immer**

**Immutable.js**

**Formik**

**React Hook Form**

**Mobx-state-tree**

**mobx-react-lite**

**Easy-peasy**

**Overmindjs**

**react-easy-state**

**Effecto**

**react-sweet-state**

**Freezer**

**Undux**

**Statusx**

**zustand**

**reatom**



# Agenda



## Recommended libraries

- Local state
- Global state
- Server state
- Immutable state
- Form state

## Deciding how to handle state

## Recommended exercises



# Demo



**To try the demos:**

- 1. Download the exercise files**
- 2. Open before**
- 3. Run “npm install”**
- 4. Import the demo in App.js**



# Local State

Goal: Manage Component State

Built into React

`useState`

Class state

`useReducer`

`refs`

Derived state in render

Also consider  
[XState](#)

Enforce state transitions  
Display transitions via state chart  
Test logic in isolation



# Global State

Goal: Share state or functions globally

Built into React  
Lift state  
Context

Also consider  
Redux  
Complex app with many state transitions  
Want to cache and share local data  
Middleware for cross-cutting concerns



# Context vs Redux

## Context

**Simple values that rarely change**

**Global state/funcs**

**Built in to React**

## VS

## Redux

**Complex values that change often**

**App state is updated frequently**

**Complex state update logic**

**Middleware**

**Medium to large codebase**

**Large team**





Home



Browse

Search...



Skill IQ



Certifications



Paths



Channels



Bookmarks



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.

# Building Applications with React 17 and Redux

by Cory House

Learn how to use React, Redux, React Router, and modern JavaScript to build an app with React. Use Webpack, Babel, Jest, React Testing Library, Enzyme, and more to build a custom React development environment and build process from the ground up.

[Resume Course](#)[Launch Sandbox](#)[Bookmark](#)[Add to Channel](#)[Download Course](#)[Schedule Reminder](#)

## Table of contents

## Description

## Transcript

## Exercise files

## Discussion

## Related Courses

This course is part of:



Building Web Applications with React Path

[Expand All](#)[Course Overview](#)

✓ 1m 36s ▾

[Intro](#)

✓ 9m 16s ▾

[Environment Build](#)

✓ 32m 45s ▾

[React Component Approaches](#)

10m 17s ▾

[Initial App Structure](#)

✓ 16m 10s ▾

[Intro to Redux](#)

18m 36s ▾

## Course info

Level Intermediate

Rating (2061)

My rating

Duration 6h 39m

Updated 18 Jul 2021

Share course

[f](#) [t](#) [in](#)

# Global State

Goal: Share state or functions globally

Built into React:

Lift state

Context

Also consider:

Redux

Complex app with many state transitions

Want to cache and share local data

Middleware for cross-cutting concerns

MobX

Optimize derived state

Manage state outside React

Recoil

Many frequently changing elements

Avoid updating unrelated parts of the UI



# Global State

Goal: Share state or functions globally

Built into React:  
Lift state  
Context

Also consider:  
**Zustand**  
Simple Redux alternative

**Jotai**  
Simple Recoil alternative

**Valtio**  
Simple Mobx alternative



# Server State

Goal: Fetch and cache server data

Built into React  
Nothing

Many use  
fetch  
Axios

We wrapped fetch in a custom hook

Also consider  
react-query  
swr  
Relay  
Apollo





# Server State: Questions to Ask

1. Should I cache data on the client for a certain period?
2. Should I load fresh data when the tab is refocused?
3. Should I load fresh data when the network reconnects?
4. Should I retry failed HTTP calls?
5. Should I return cached data, then fetch fresh data behind the scenes?
6. Should I handle server cache separately from app state?
7. Should I avoid refetching recently fetched data?
8. Should I prefetch data the user is likely to want?

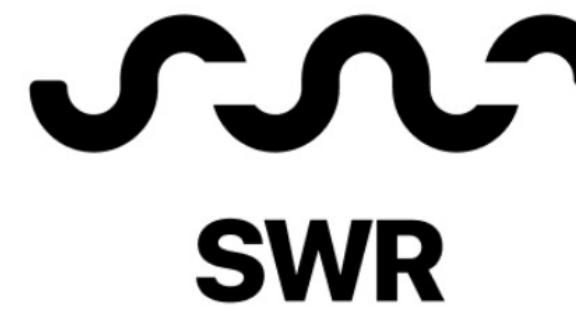


# Server State

Goal: Fetch and cache server data



**react-query**



**swr**

**Any endpoint**



**Relay**

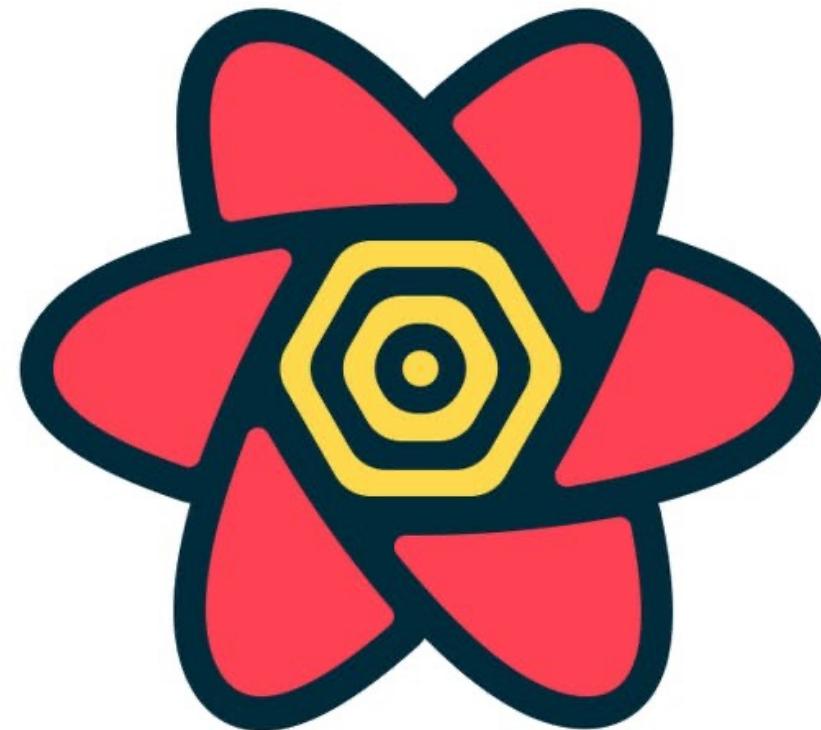


**Apollo**

**GraphQL focused**



# react-query and swr



SWR

**Automatic, dedicated server cache**

- Separate server-cache
- Invalidates and refetches stale data

**Dedupes requests**

**Auto retries**

**Refetch on refocus / network reconnect**



# **stale-while-revalidate**



I'm okay with showing slightly stale data for a moment

- 1. Display cached record**
- 2. Request fresh record behind the scenes**
- 3. Display the fresh record if it's newer**



# Immutable State

Goal: Enforce immutability

Built into React:  
Nothing  
We used plain JS

Also consider:  
[Immer](#)  
Write “mutative” code  
Plain JS  
Small and easy to learn  
Auto freezes



# Immer Example

```
import produce from "immer"
const user = {
  name: "Cory",
};

const userCopy = produce(user, draft => {
  draft.name= "Cory2"
})

console.log(user.name); // Cory
console.log(userCopy.name) // Cory2
```



# Form State

Goal: Manage Form State

**Built into React:**  
**State**  
**Event handlers**  
**Derived state**

**Also consider:**  
**Formik**  
**React Hook Form**  
Reduce form boilerplate  
Enforce conventions





FORMIK docs

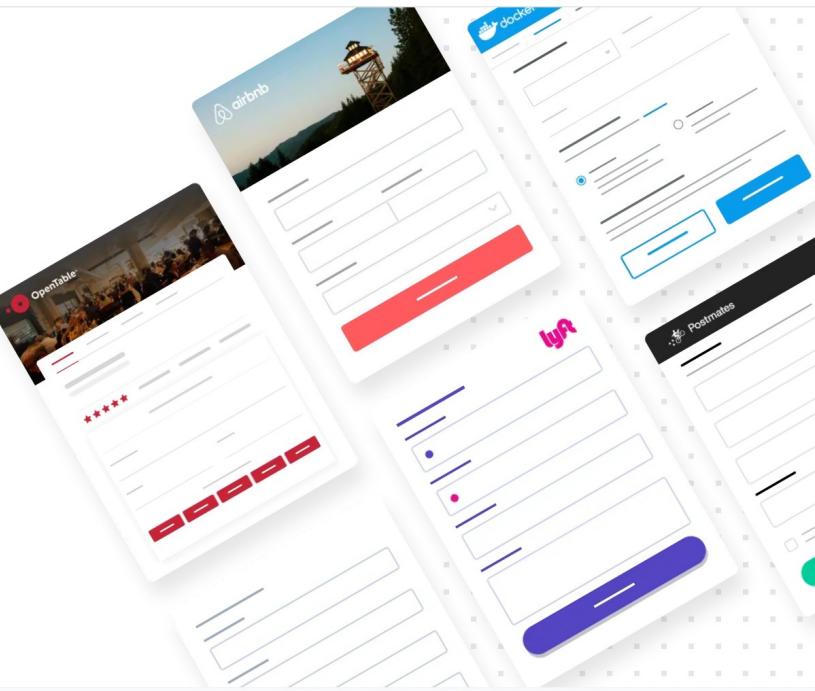
Docs Blog Users Feedback

## Build forms in React, without the tears

Formik is the world's most popular open source form library for React and React Native.

Get Started

[GitHub](#)



### Declarative

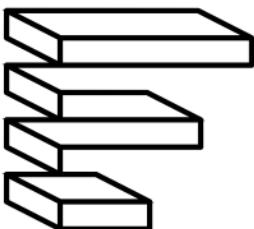
Formik takes care of the repetitive and annoying stuff—keeping track of values/errors/visited fields, orchestrating validation, and handling submission—so you don't have to. This means you spend less time wiring up

### Intuitive

No fancy subscriptions or observables under the hood, just plain React state and props. By staying within the core React framework and away from magic, Formik makes debugging, testing, and reasoning about your forms a breeze. If

### Adoptable

Since form state is inherently local and ephemeral, Formik does not use external state management libraries like Redux or MobX. This also makes Formik easy to adopt incrementally and keeps bundle size to a minimum.



# Formik

Q Twitter

Home Get Started API TS Advanced FAQs Tools ▾ Resources Releases

English 13,336



## React Hook Form

Performant, flexible and extensible forms with easy-to-use validation.

Demo

Get Started ▶

```
my-app-2 [~/git/my-app-2] - .../src/App.js
1 import React from "react";
2
3 let renderCount = 0;
4
5 function App() {
6   renderCount++;
7
8   const onSubmit = data => console.log(data);
9
10  return (
11    <form onSubmit={onSubmit}>
12      <label>First Name</label>
13      <input name="firstName" />
14
15      <label>Last Name</label>
16      <input name="lastName" />
17
18      <p>Render counter: {renderCount}</p>
19
20      <button>Submit</button>
21    </form>
22  );
23}
```



# React Hook Form



# Eight Ways to Handle State in React Apps



**URL**



**Web storage**



**Local state**



**Lifted state**



**Derived state**



**Refs**



**Context**



**Third party library**

## When to use it

**Sharable app location**

**Persist between sessions, one browser**

**Only one component needs the state**

**A few related components need the state**

**State can be derived from existing state**

**DOM reference, state that isn't rendered**

**Global or subtree state**

**Global state, Server state, Form state, etc.**



# Deciding How to Handle State

1. Does it belong in the URL? (current page, current record, sorting, scroll location...)  
**Keep URL-related state in the URL.**
2. Want to persist data across sessions or make data available offline?  
**Consider web storage (localStorage, IndexedDB, etc)**
3. Is it server data?  
**Try react-query or swr. Using GraphQL? Then also consider Relay / Apollo.**
4. Is it a DOM element reference, state that doesn't change, or not rendered at all?  
**Use a ref.**
5. Can it be derived from existing props, state, URL, etc?  
**Derive it “on-the-fly” as part of each render (memoize if expensive).**
6. Does only one component use the data?  
**Use local state.**
7. Do a few related components use it?  
**Store state in a common parent.**
8. Is it global state? Consider, *in order*.  
**Store in App's root component, context, or separate library like Redux.**





# Exercises

**Add backpacks to the store**

**Finish the checkout process**

- Accept billing info and payment
- Save partially completed checkout
- Display final order confirmation

**Display cart quantity in nav**

**Deduplicate requests in useFetchAll**

**Add init arg to useFetch hook**

**Try caching via react-query or swr**



# Thanks for watching!



**Cory House**   
Pluralsight Author

 Following

6778 Followers

Cory is the principal consultant at reactjsconsulting.com, where he has helped dozens of companies transition to React. Cory has trained over 10,000 software developers at events and businesses worldwide. He is a seven time Microsoft MVP, and speaks regularly at conferences around the world. Cory...

Show more...

 [www.bitnative.com](http://www.bitnative.com)  
 [Twitter](#)  
 [LinkedIn](#)

CONTENT AUTHORED

**14**

All time

TOPICS AUTHORED



TOTAL RATINGS

**11,446**

AVG CONTENT RATING

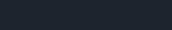
**4.7**

## Content authored

### React 18: The Big Picture

Course · Beginner · 1 hr 6m · Dec 19, 2022 ·  (36)

### Next.js 12: The Big Picture

Course · Beginner · 50m · Jun 23, 2022 ·  (44)

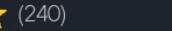
### Building Applications with React 17 and Redux

Course · Intermediate · 6 hr 39m · Jul 18, 2021 ·  (2,061)

### Building a JavaScript Development Environment

Course · Beginner · 4 hr 55m · Mar 15, 2021 ·  (736)

### Managing React State

Course · Intermediate · 5 hr 6m · Aug 19, 2020 ·  (240)

### React: The Big Picture

Course · Beginner · 1 hr 10m · May 10, 2020 ·  (968)

### Clean Coding Principles in C# RETIRED

Course · Beginner · 3 hr 19m · Jan 1, 2020 ·  (482)

### Building Applications with React 16 and Flux 3 RETIRED

Course · Intermediate · 5 hr 11m · Jun 18, 2019 ·  (1,682)

### Securing React 16 Apps with Auth0

Course · Intermediate · 3 hr 18m · Nov 29, 2018 ·  (174)

# Summary



## Global state

- Redux, Recoil

## Finite State

- XState

## Server state

- react-query, swr, Relay, Apollo

## Form state

- Formik, react-hook-form

## Immutable state

- Immer, Immutable

