Managing State via 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
Effector
react-sweet-state
Freezer
Undux
Statux
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 11/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

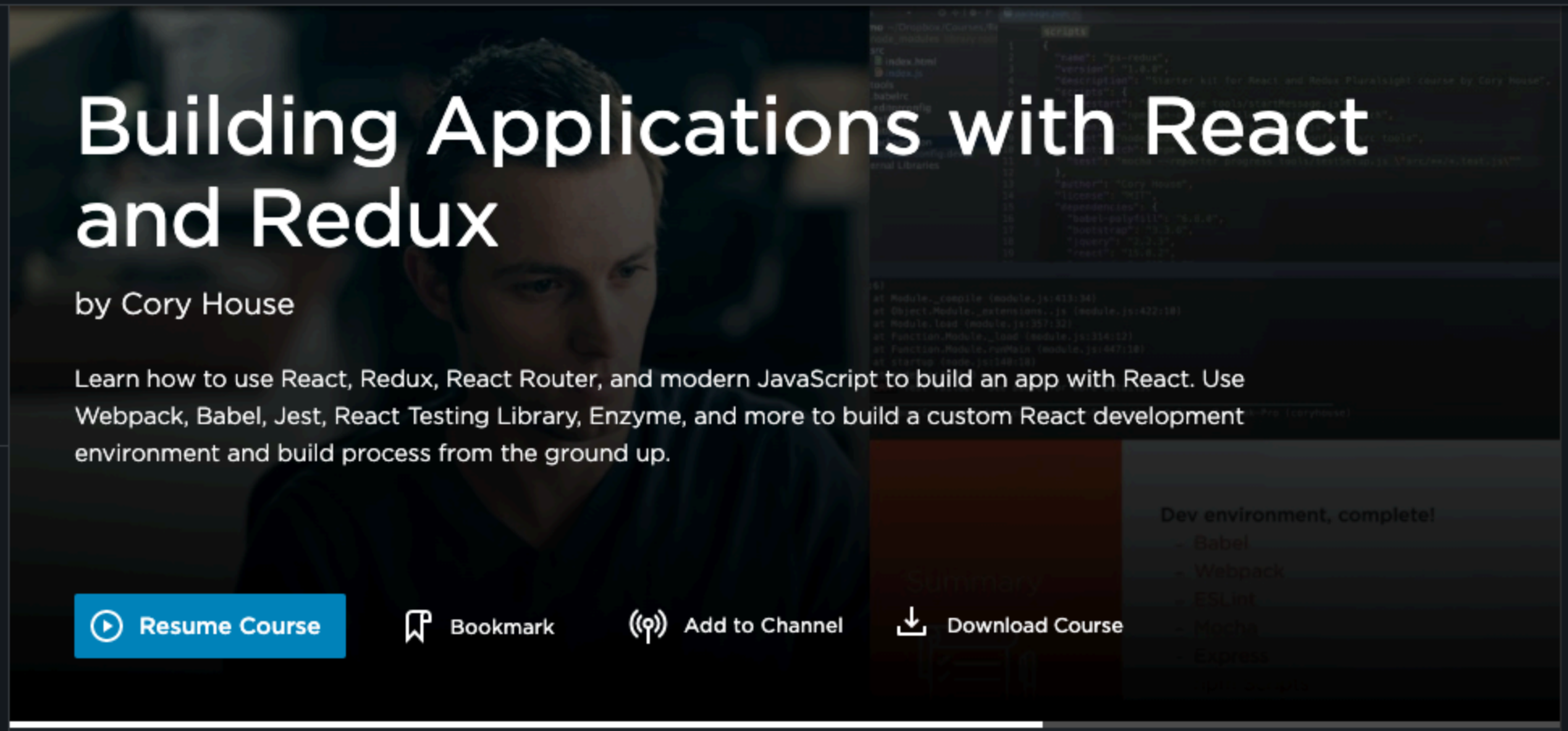
Also consider:

Redux

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

Context vs Redux

Context	Redux
Simple values that rarely change	Complex values that change often
Global state/funcs	App state is updated frequently
Built in to React	Complex state update logic
	Middleware for cross-cutting concerns
	Medium to large codebase
	Large team



Building Applications with React 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 | Bookmark | Add to Channel | Download Course

Table of contents | Description | Transcript | Exercise files | Discussion | Learning Check | Related Courses

This course is part of: React Path

Expand All

Course Overview ✓ 1m 36s

Course author

Cory House

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...

Course info

Level	Intermediate
Rating	★★★★★ (1572)
My rating	★★★★★
Duration	6h 39m
Updated	12 Mar 2019

Share course

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 (beta)

- Many frequently changing elements
- Avoid updating unrelated parts of the UI

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

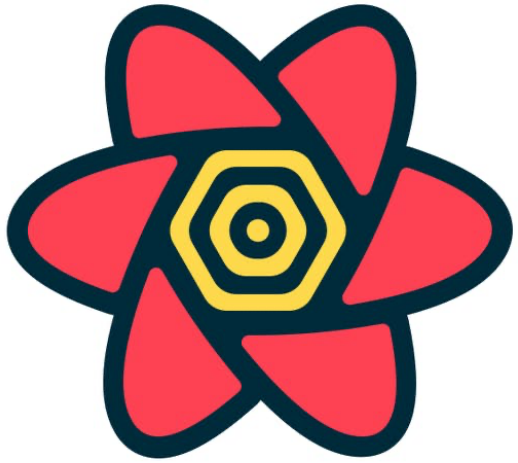
Apollo

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



APOLLO

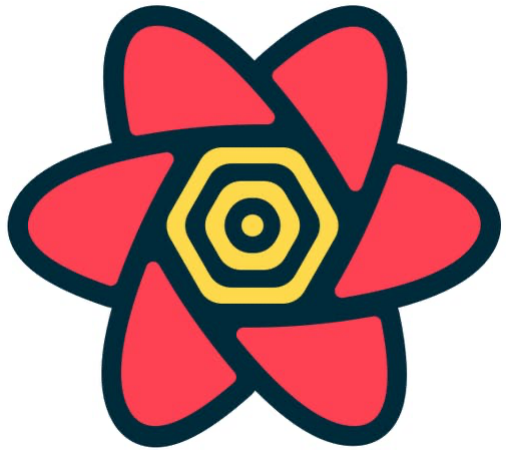
react-query swr

Relay Apollo

Any endpoint

GraphQL focused

react-query and 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

React Hook Form

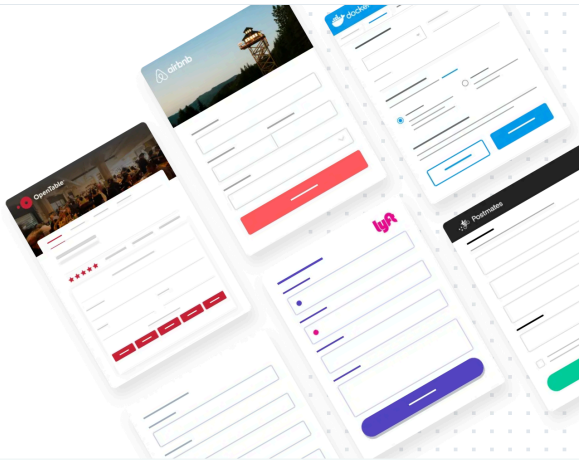
FORMIK docs

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.

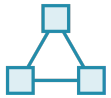
Home Get Started API TS Advanced FAQs Tools Resources Releases English 13,336

React Hook Form

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

Demo Get Started ▶

Eight Ways to Handle State in React Apps

	When to use it
 URL	Sharable app location
 Web storage	Persist between sessions, one browser
 Local state	Only one component needs the state
 Lifted state	A few related components need the state
 Derived state	State can be derived from existing state
 Refs	DOM reference, state that isn't rendered
 Context	Global or subtree state
 Third party library	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!

@housecor

Cory House
Pluralsight Author

Following

4543 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
Twitter
LinkedIn

CONTENT AUTHORED	TOPICS AUTHORED	TOTAL RATINGS	AVG CONTENT RATING
11 All time	javascript	9,344	4.7

Content authored

React: The Big Picture NEW!
Course · Beginner · 1 hr 10m · May 10, 2020 · ★★★★½ (544)

Clean Coding Principles in C#
Course · Beginner · 3 hr 19m · Jan 1, 2020 · ★★★★★ (120)

Building Applications with React and Flux
Course · Intermediate · 5 hr 11m · Jun 18, 2019 · ★★★★★ (1,465)

Building Applications with React and Redux
Course · Intermediate · 6 hr 39m · Mar 11, 2019 · ★★★★★ (1,574)

Securing React Apps with Auth0
Course · Intermediate · 3 hr 18m · Nov 29, 2018 · ★★★★★ (104)

Creating Reusable React Components
Course · Intermediate · 6 hr 20m · Jun 4, 2017 · ★★★★★ (116)

Building a JavaScript Development Environment
Course · Beginner · 5 hr 19m · Nov 9, 2016 · ★★★★★ (644)

HTML5 Web Component Fundamentals
Course · Beginner · 5 hr 3m · Jan 8, 2015 · ★★★★½ (456)

Becoming an Outlier: Reprogramming the Developer Mind
Course · Intermediate · 2 hr 33m · Apr 23, 2014 · ★★★★½ (817)

Architecting Applications for the Real World in .NET
Course · Intermediate · 2 hr 52m · Jan 6, 2014 · ★★★★½ (1,629)

Courses by Cory

Summary

State Libraries

Global state

- Redux, Recoil

Finite State

- XState

Server state

- react-query, swr, Relay, Apollo

Form state

- Formik, react-hook-form

Immutable state

- Immer, Immutable