Using More Hooks: useContext, useReducer, useCallback, and useMemo

Peter Kellner
DEVELOPER, CONSULTANT AND AUTHOR

@pkellner linkedin.com/in/peterkellner99 PeterKellner.net

Learned in Past Module

useState

useRef

useEffect

More React Hooks Coming Up

useContext

useReducer

useCallback

useMemo

React Hooks Covered so Far

useContext

useReducer

useCallback

useMemo

React Hooks Covered

useContext

useReducer

useCallback

useMemo

Declaratively create functionality.

Is there new functionality in React Hooks?

Is there new functionality in React Hooks?

No

React Hooks are here
to stay.

Are React class components dead?

Are React class components dead?

No

Should you update all your Class Components?

Should you update all your Class Components?

No

We will learn React Hooks
from the ground up.

React Hooks usage rules coming up.

Clear, composable, and declarative way.

React Hooks makes developing apps easier and with less complexity.

What Makes React Hooks Great

Composition instead of inheritance

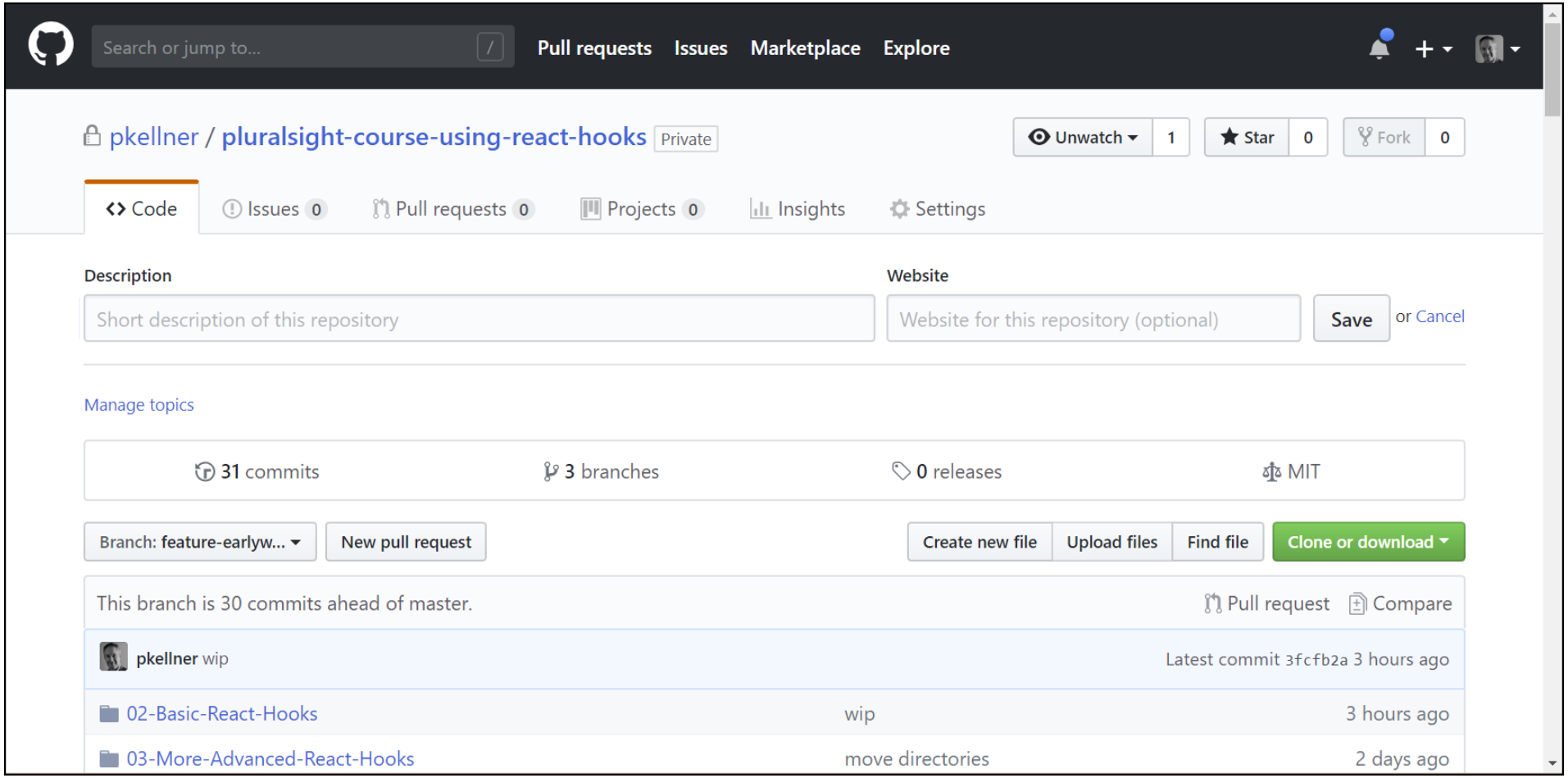
Best parts of mixins

Replace prop drilling, render props, HOCs

State helps build highly performant web apps.

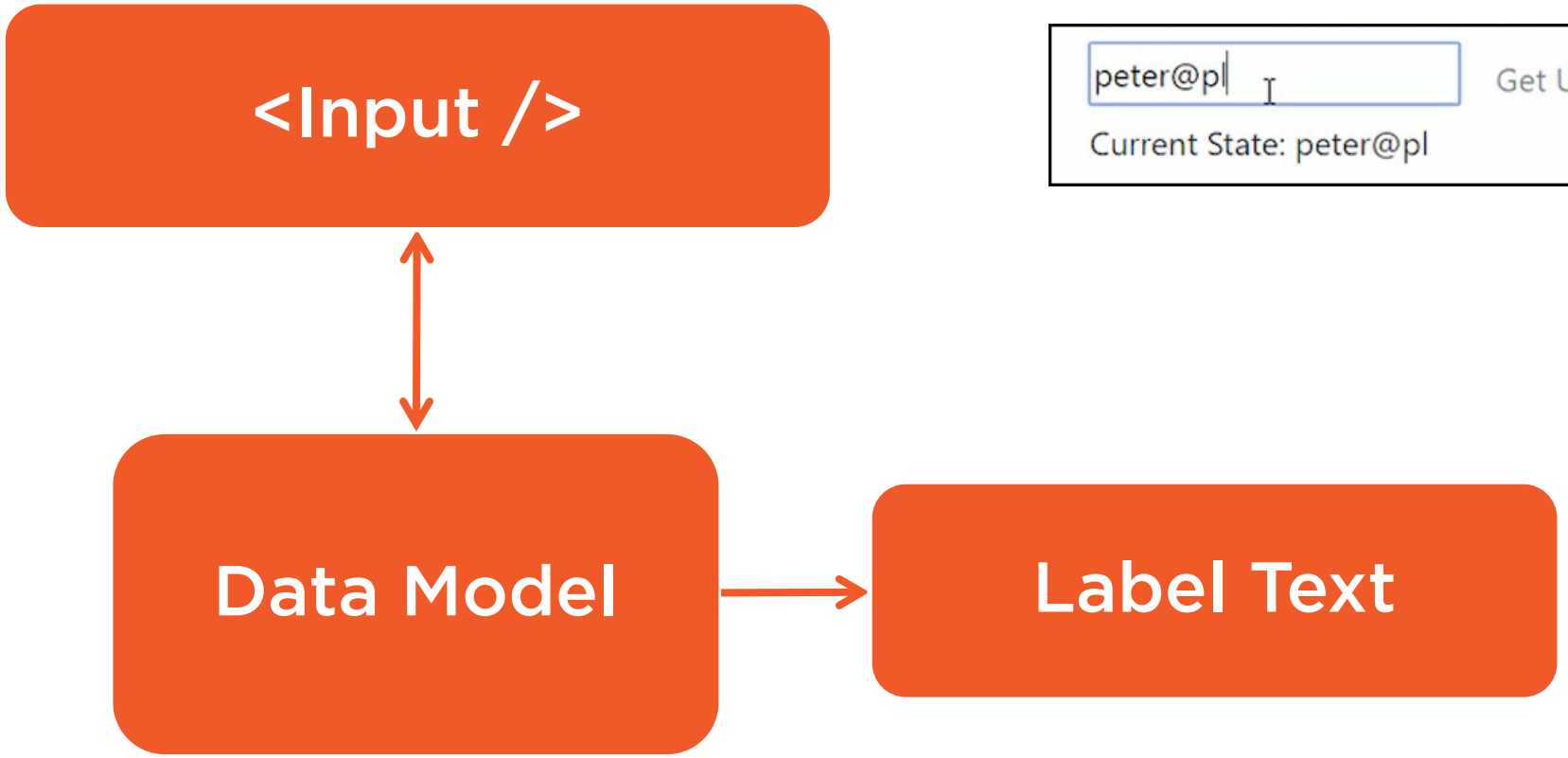
The Course Source Code on GitHub

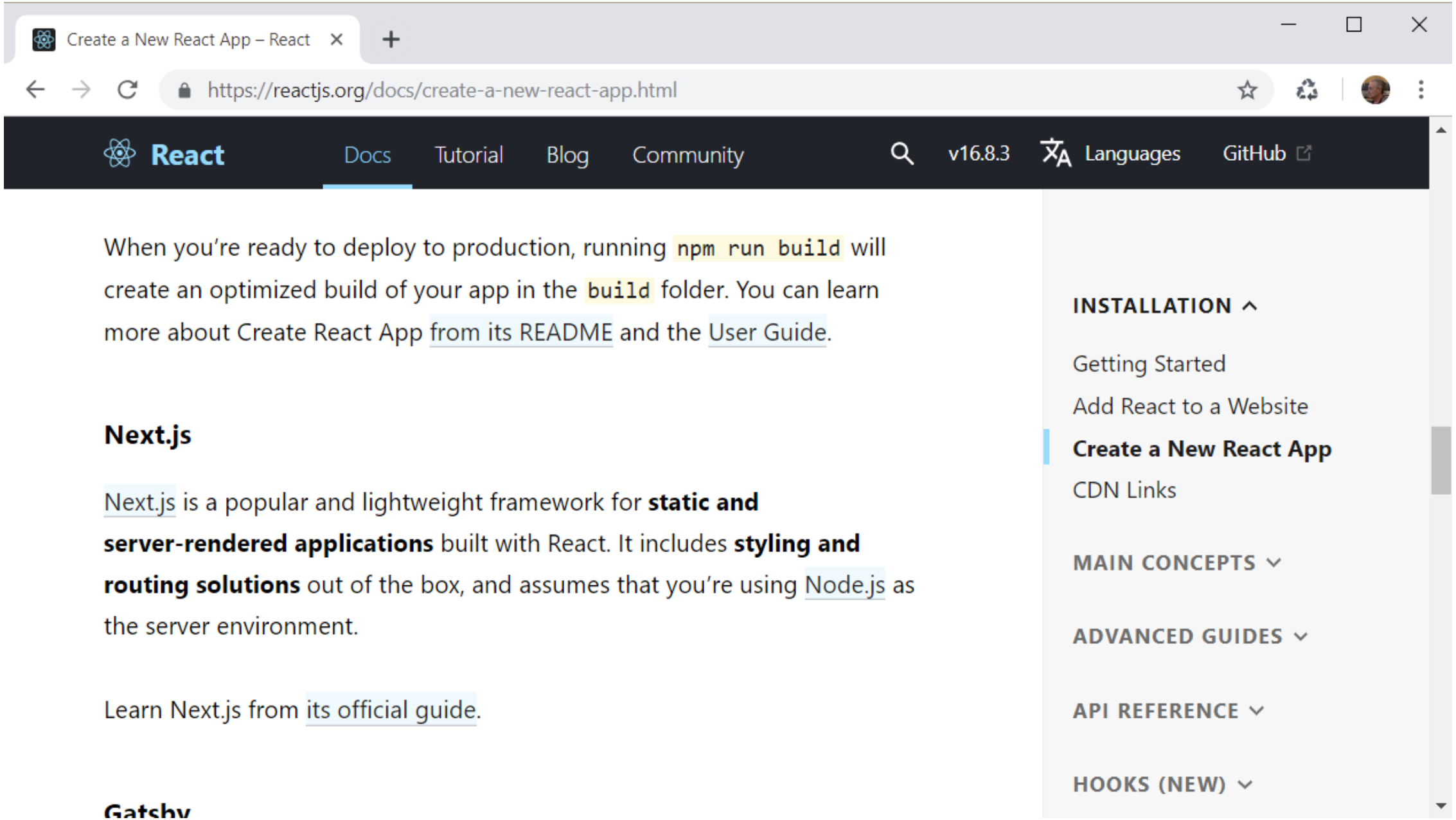
github.com/pkellner/pluralsight-course-using-react-hooks



Two-way Data Binding

Two-way Data Binding




Create a New React App – React
https://reactjs.org/docs/create-a-new-react-app.html
React
Docs
Tutorial
Blog
Community
v16.8.3
Languages
GitHub
When you're ready to deploy to production, running npm run build will create an optimized build of your app in the build folder. You can learn more about Create React App from its README and the User Guide.
Next.js
Next.js is a popular and lightweight framework for static and server-rendered applications built with React. It includes styling and routing solutions out of the box, and assumes that you're using Node.js as the server environment.
Learn Next.js from its official guide.
Gatsby
INSTALLATION
Getting Started
Add React to a Website
Create a New React App
CDN Links
MAIN CONCEPTS
ADVANCED GUIDES
API REFERENCE
HOOKS (NEW)

State helps build highly performant web apps.

useRef

Primarily used to allow access directly to an element in the DOM.

Side effects === bad thing.

Pure functional component with no side effects is also called “pure”.

React Hook useEffect causes side effects.

Adding and removing DOM listeners is great example.

Pure component guarantees no side effect, non pure may also have none.

Why bother with more than the three basic hooks?

Improved performance!

Global configuration support.

Cleaner code.

Prop drilling, HOCs anti-patterns.

useContext saves the day.

useContext feels like using mixins.

Are reducers a good thing or bad thing?

useReducer makes
programming reducers fun!

The React Hook useState is extremely powerful.

useState is built with
useReducer under it.

Reducer Defined

```
(previousState, action) => newState
```