

React Day Berlin 2018

Takeaways sharing

kcliu

React Day Berlin

Build code, not walls

November 30, 2018

Berlin, Germany

React Day Berlin

- 800 React fans and lovers
- 24+ Speakers
- 2 Tracks (General + Native/Reason)

[WATCH LIVE STREAM](#)



Introduction

- Conference
 - 800 React fans, 2 tracks
 - West Wall (React)
 - East Wall (React Native)
- Workshops
 - React
 - GraphQL
 - ReasonML
- Videos already online
 - Search react berlin on YouTube

The New Hotness: An Exploration of Reacts Exciting New Features

- By Ken Wheeler
- What's new in React 16.6, 16.7
- React hooks
 - A way to write function based components with class component capabilities
 - Built-in hooks
 - `useState()`, `useEffect()`, `useRef()`, `useContext()` ...
 - custom hooks
 - Why hooks
 - future optimizations
 - Less code
 - More readable, better organization
- Detailed Demo code tracing

- More readable, better organization



Feature Driven Architecture

- By Oleg Isonen
- Goals:
 - Discoverability
 - Work parallelisation
 - Controlling shared abstractions
 - Refactoring
 - AB Tests
 - Integration Tests

Feature Driven Architecture

- A set of principles that helps you to define the boundaries.
- Principles
 - Decentralization
 - Explicit sharing
 - Co-location
 - Decoupling
 - Disposability

Decentralization

- Avoid code base heavily interconnected
- Match UI

1. Decentralization

Organise by **features** instead of types.

```
src/  
└─ features/  
   └─ featureA/  
   └─ featureB/  
   └─ featureC/
```


Explicit Sharing

To share or not to share?

- Needed in at least 2 places
- Non-trivial logic
- Low frequency of change

Co-location

- Navigating the Giant
 - Better finding your files
- About DX
- Code structure
 - example: hooks
- Folder structure
 - example: ducks pattern

3. Co-location

Directory structure

```
src/  
├── shared/  
│   ├── components/  
│   └── api/  
└── features/  
    ├── {feature}/  
    │   ├── containers/  
    │   ├── actionCreators/  
    │   ├── actionTypes/  
    │   ├── renderers/  
    │   ├── selectors/  
    └── reducer
```

Decoupling & isolation

- A feature should not depend on other features
 - A page should not depend on other pages
 - A shared abstraction should not depend on either.
-
- Example: header with login information into props
 - Example: scoped state, renderProps, Context API

Disposability

- Optimize for ease of removal
- Example: remove FeatureA/

Tip Top Testing in JavaScript

- By Jack Franklin
- Generic tips on how to write good tests.
- online courses about testing



If your tests can fail without
any of the code you're testing
changing, your test is not
properly *isolated.*



Option 1:

```
processUserClickOnItem({ id: 123 })  
expect(frontendEventsLog[frontendEventsLog.length - 1]).toEqual({  
  type: 'item_click',  
  data: { item_id: item.id },  
})
```

Option 2:

```
processUserClickOnItem({ id: 123 })  
expect(threadEventsLogger.log).toHaveBeenCalledWith('item_click', {  
  item_id: 123,  
})
```

Tip Top Testing in JavaScript

- Many test frameworks will pass a test without an assertion!
- If you write a test and it passes first time, try to break it



Write code

wait 5 seconds for webpack

manually refresh the browser

click the button you're working on

go back to your editor

Check if it worked



Write code

hit save in your editor

Run tests



*Test things by
calling them just
like you do in real
life*

Example

```
it('filters the items to only shirts', () => {  
  const shirtFinder = new ShirtFinder({ priceMax: 5000 })  
  expect(shirtFinder.__foundShirts).toEqual([])  
  expect(shirtFinder.getShirts()).toEqual([])  
})
```

Test in React

**So test your
components** *as if
you are a user.*



Which test is better?

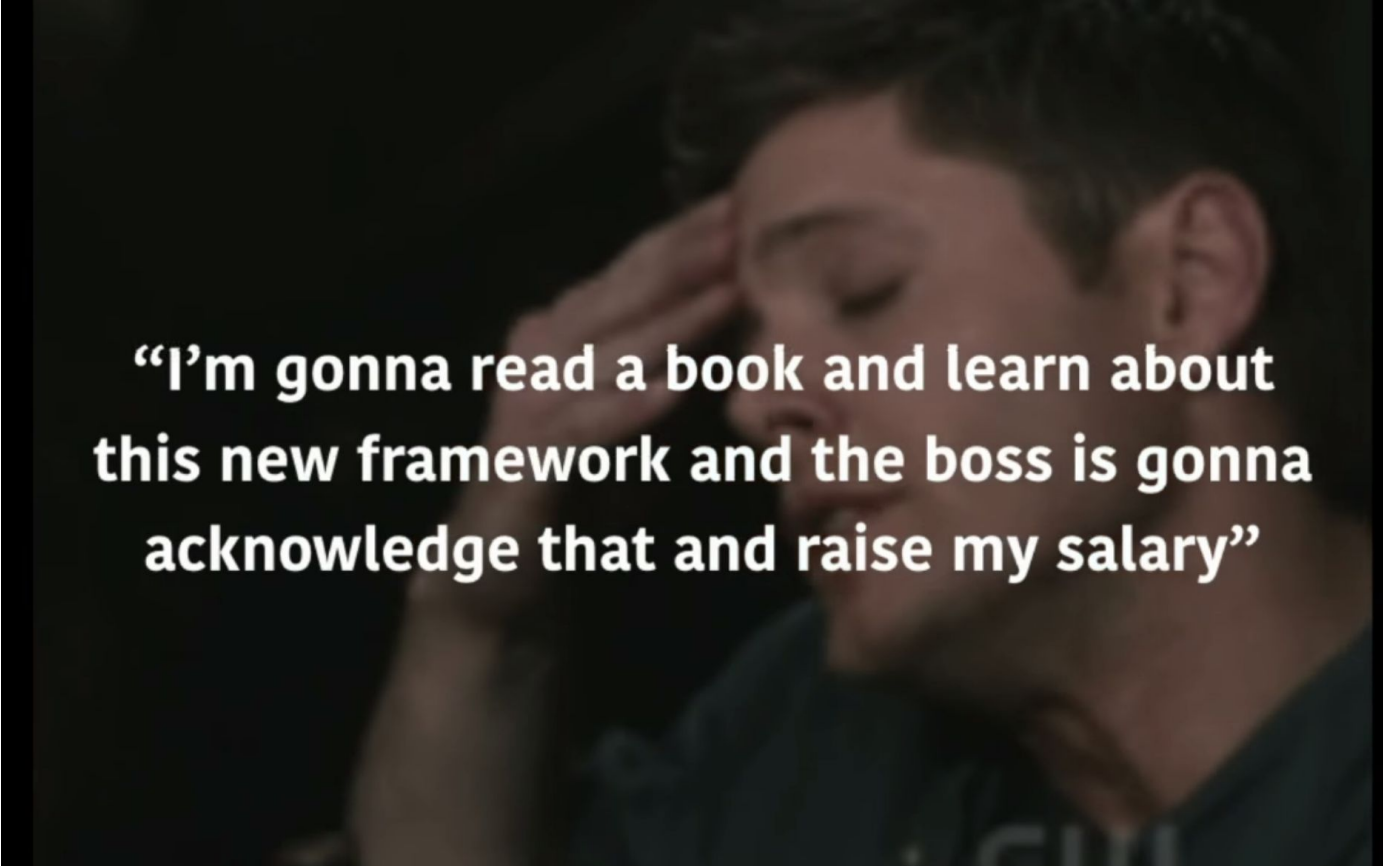
```
const wrapper = mount(<Button />)
wrapper.find('a').simulate('click')
expect(wrapper.getState().isDisabled).toEqual(true)
```

Or:


```
const wrapper = mount(<Button />)
wrapper.find('button').simulate('click')
expect(wrapper.find('button').prop('disabled')).toEqual(true)
```

Navigating the Hype-Driven Frontend Development World Without Going Insane

- By Kitze

A man with dark hair and a beard is shown in profile, looking down with his hand pressed against his forehead in a gesture of stress or frustration. The background is dark and out of focus.

**“I’m gonna read a book and learn about
this new framework and the boss is gonna
acknowledge that and raise my salary”**






**You studied for 15 years
but somewhere out there
there's a nerdy teen
who learned Vue.js in 2 months
on Udemy and made a
Tinder For Cats app that's making
more money than you're ever
gonna make in your entire career**

 How **easy** is the transition?

 Is **now** the right time?

 Do I **really** need it?

floats  flexbox

 easy  right time  really need it

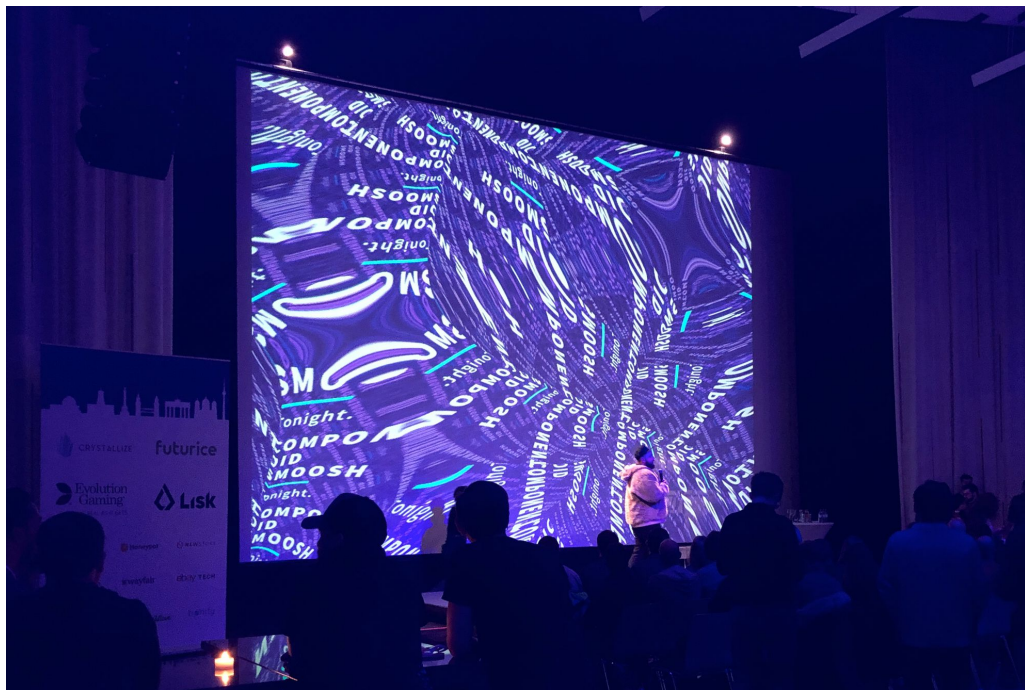
- Don't get yourself to the Fear Of Missing Out. It's just a side-effect of wrong priorities.

The Lonely and Dark Road to Styling in React

- By Sara Vieira
- Plain old CSS and BEM
- Preprocessors
- CSS modules
- CSS in JS

Comparison

- Conf features
 - International
 - Sponsors
 - workshop
 - Professional live streaming
 - No Q&A
- Networking
 - Pre-party
 - After party
 - Lunch time





📌 Pinned by React Day Berlin

React Day Berlin 1 month ago

6:10 Opening

→ 20:04 Keynote - Ken Wheeler

1:02:10 Forbes

→ 1:25:15 Oleg

2:25:10 Monica

→ 2:50:51 Jack

3:18:21 Max

3:24:30 Jamie

3:31:59 Bard

3:37:15 Denis

3:43:48 Denys

4:51:30 GitNation OS Stage

5:21:43 Jani - Poem to JavaScript

→ 5:25:50 Kitze

5:59:30 Raphael

6:26:53 Rebecca

7:24:00 Alex

→ 7:52:00 Sara