

React Hooks in Reason

Who's this guy?

Messenger Web, Reason,
ReasonReact, internal
infrastructure at FB

New to NYC!

@rickyvetter everywhere



What's a hook?

- `createClass`
 - Not sexy
 - Comes with runtime
 - Hard to control state updates
- Function component
 - Sexy
 - Can't be stateful
- Class components
 - Kinda sexy
 - Lots of unintuitive binding issues. Hard to control state updates

What's a hook?

What if we make functions stateful?

What's a hook?

```
import React, { useState } from 'react';

function Example() {
  // Declare a new state variable, which we'll call "count"
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>
        Click me
      </button>
    </div>
  );
}
```

What's a hook?

- createClass
 - Not sexy
 - Comes with runtime
 - Hard to control state updates
- Function component
 - Sexy
 - ~~Can't~~ Can be stateful
 - Group logic by effects instead of by point of time in lifecycle
- Class components
 - Kinda sexy
 - Lots of unintuitive binding issues. Hard to control state updates

Use in Reason

- Bindings for hooks are pretty easy
 - We already have most of them! But - you can only use them in function components.
- We need to match React api - specifically `React.createElement(component, object)`, where component takes object and returns a React element
- This means ... we need to bind to objects

Object Binding Strategies

- Js.t {"foo": int, "bar": option(string)}
 - Doesn't support optional fields - {"foo": int}

Object Binding Strategies

- `bs.deriving abstract`
 - Supports optional fields!
 - no destructuring, no default arguments, lots of shadowing
 - ```
let foo = switch (props->foo) {
 | Some(foo) => foo
 | None => 1
};
```

# Object Binding Strategies

- `bs.obj`
  - Supports optional fields!
  - no destructuring, no default arguments

# Object Binding Strategies

- `bs.obj + decorator`
  - Supports optional fields!
  - can recover destructuring, defaults

# Show Code Here

# Upgrade Path

# Safety

**Brisk/Revery**

# Thanks!

[github.com/rickyvetter/ocaml-nyc-2019-02](https://github.com/rickyvetter/ocaml-nyc-2019-02)

<https://reactjs.org/docs/hooks-overview.html>

<https://reactjs.org/docs/hooks-reference.html>