

React & Flow

*Corey Grunewald
Senior Software Engineer @ Netflix*



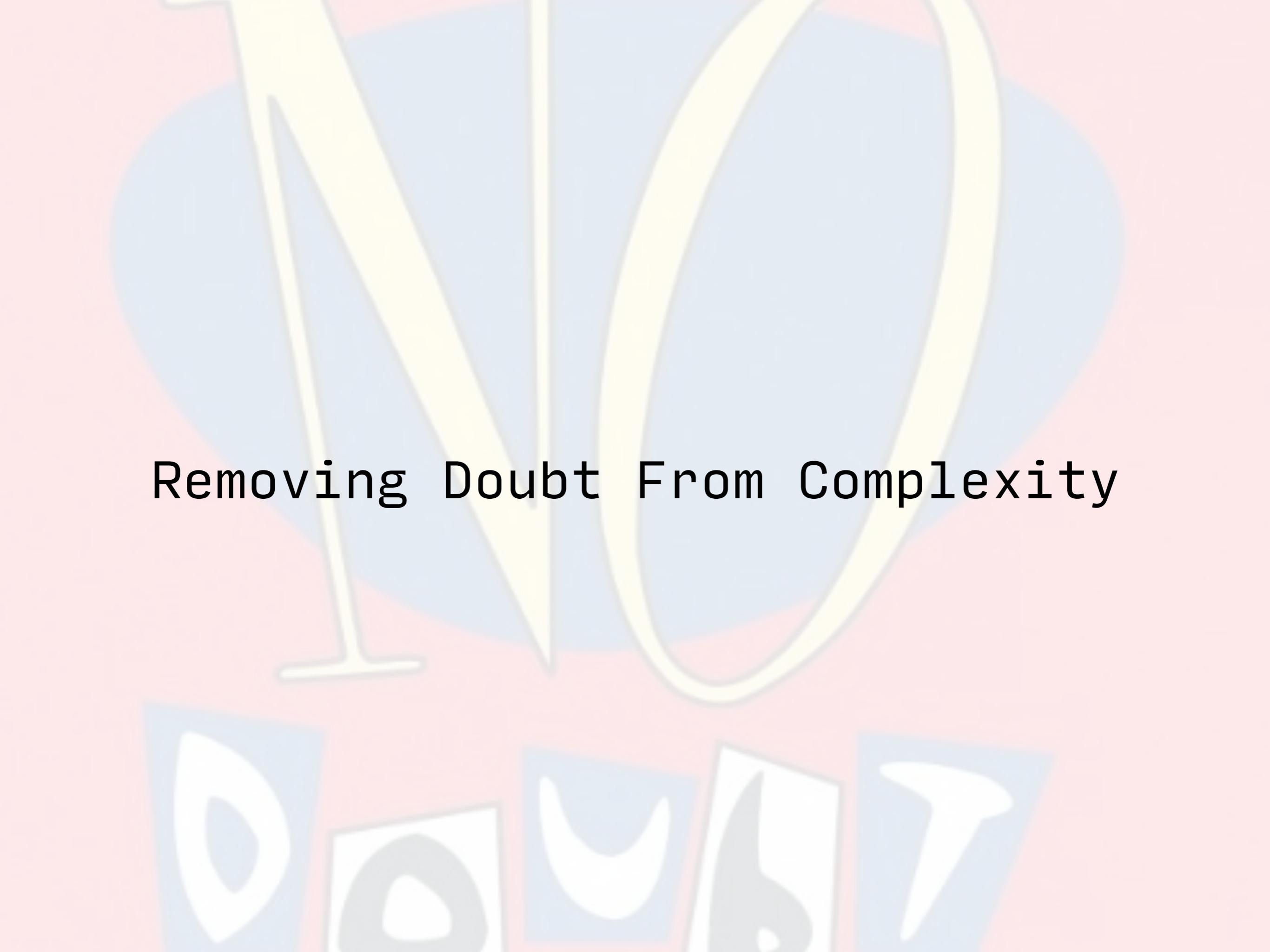
“Successful Production Push”



“Whoops”



“Try, try again”



Removing Doubt From Complexity

Known Knowns vs. Unknown Unknowns

“But we have unit tests!?”

“and jsdoc...”

We can do better.

Flow-ed in 60 Seconds

```
function logKibble(pieces) {  
  return 'There are ${pieces.length} pieces of Kibble left.';  
}  
// later in your application...  
logKibble({ kibble: [1, 2, 3] });
```

src/views/shelter.jsx:194

194: return 'There are \${pieces.length} of Kibble left.';
 ^^^^^^ **property 'length'.**

Property not found in

194: return 'There are \${pieces.length} of Kibble left.';
 ^^^^^^ **object literal**

Flow-ed in 60 Seconds

```
type Kibble = Array<any> | number;

function logKibble(pieces: Kibble): string {
  const count = Array.isArray(pieces)
    ? pieces.length
    : pieces;

  return `There are ${count} pieces of Kibble left.`;
}

const one = logKibble([1, 2, 3]);
const two = logKibble(8);
const weeklyKibble = (one + two) * 7;

src/views/shelter.jsx:203
203: const weeklyKibble = (one + two) * 7;
                                         ^^^^^^ string. This type is incompatible with
203: const weeklyKibble = (one + two) * 7;
                                         ^^^^^^ number
```

Case Study



Snoopy

[Adopt Me!](#)
Located in: Los Gatos

From: Charlie



Garfield

[Adopt Me!](#)
Located in: Los Gatos

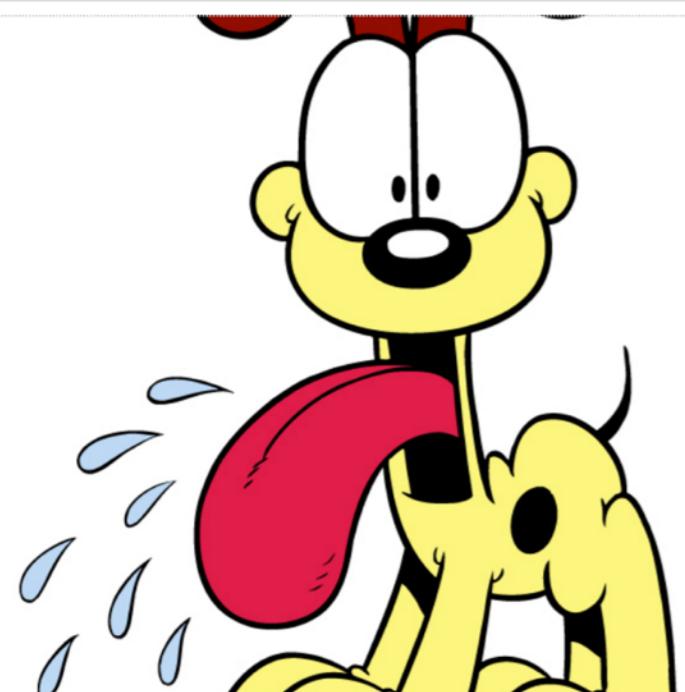
From: John



Salem

[Adopt Me!](#)
Located in: Los Gatos

From: Sabrina



Odie

[Adopt Me!](#)
Located in: Los Gatos

From: John



Scooby

[Foster Me!](#)
Located in: Santa Cruz

From: Shaggy



Pluto

[Foster Me!](#)
Located in: Santa Cruz

From: Walt



Snoopy

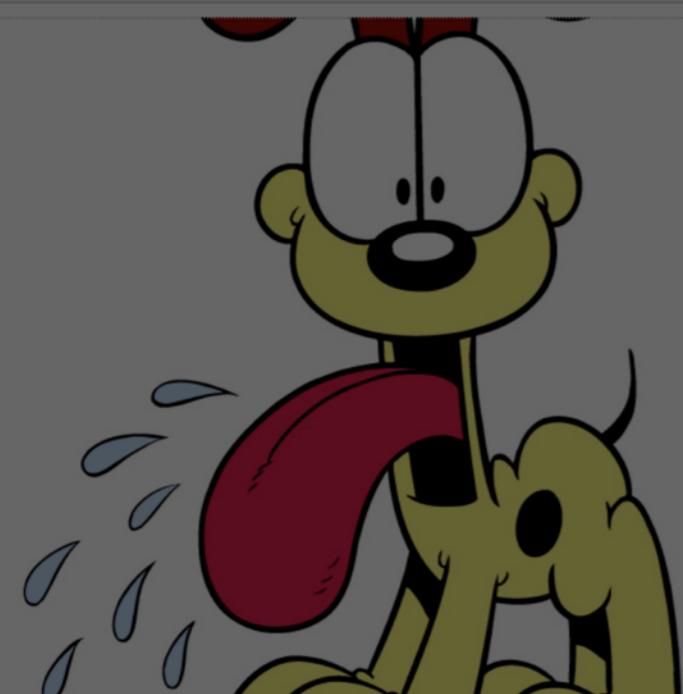
[Adopt Me!](#)
Located in: Los Gatos

From: Ch



Snoopy

From: Charlie

 Name Email Zip

Odie

[Adopt Me!](#)

Located in: Los Gatos

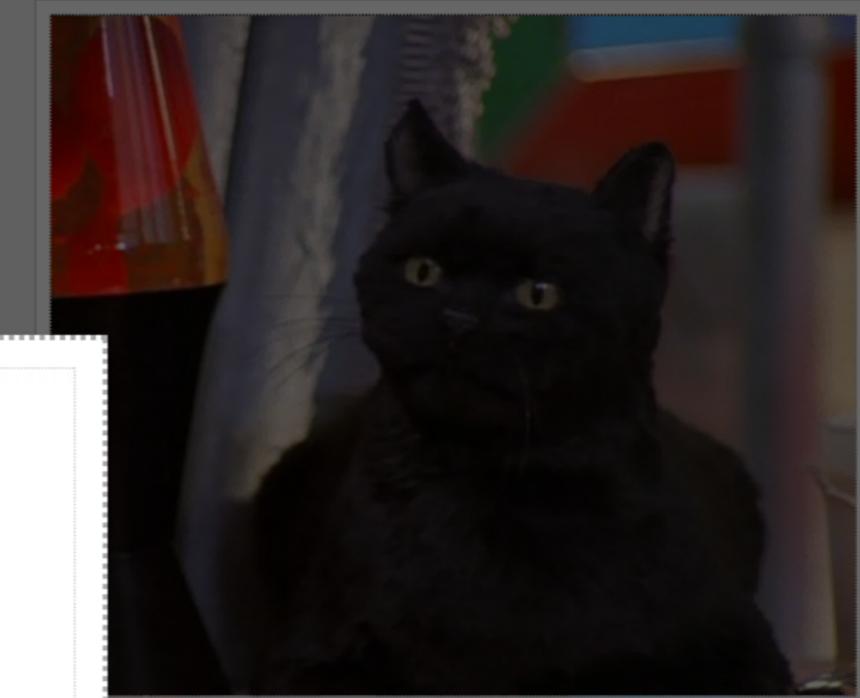
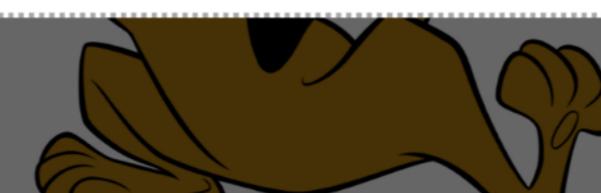
From: John

Scooby

[Foster Me!](#)

Located in: Santa Cruz

From: Shaggy



n

[Adopt Me!](#)
Located in: Los Gatos

From: Sabrina



Pluto

[Foster Me!](#)

Located in: Santa Cruz

From: Walt

```
declare type PetInquiry = {
  name: string;
  email: string;
  zip: number;
}

declare type ActionSendInquiry = {
  type: 'SEND_INQUIRY';
  action: PetAction;
  id: number;
  inquiry: PetInquiry;
}

declare type SendInquiry =
(x: ActionSendInquiry) => void;

// @flow

type ModalProps = {
  sendInquiry: SendInquiry;
  pet: Pet;
};

type ModalState = { inquiry: PetInquiry };

class PetModal extends React.Component {

  props: ModalProps;
  state: ModalState;
  onSubmitClick: () => void;

  constructor(props: ModalProps) {
    super(props);

    this.state = { inquiry: {} };

    this.onSubmitClick = () => {
      props.sendInquiry({
        type: 'SEND_INQUIRY',
        action: props.pet.action || 'adopt',
        id: props.pet.id,
        inquiry: this.state.inquiry
      });
    };
  }

  render() {...}
}
```

```
src/views/modal-example.jsx:10
10:     inquiry: PetInquiry;
          ^^^^^^^^^^ property 'email'. Property not found in
22:     this.state = { inquiry: {} };
          ^& object literal
```

```
src/views/modal-example.jsx:10
10:     inquiry: PetInquiry;
          ^^^^^^^^^^ property 'name'. Property not found in
22:     this.state = { inquiry: {} };
          ^& object literal
          ^& object literal
```

```
src/views/modal-example.jsx:10
10:     inquiry: PetInquiry;
          ^^^^^^^^^^ property 'zip'. Property not found in
22:     this.state = { inquiry: {} };
          ^& object literal
```

```
declare type PetInquiry = {
  name: string;
  email: string;
  zip: number;
}

declare type ActionSendInquiry = {
  type: 'SEND_INQUIRY';
  action: PetAction;
  id: number;
  inquiry: PetInquiry;
}

declare type SendInquiry =
(x: ActionSendInquiry) => void;

// @flow

type ModalProps = {
  sendInquiry: SendInquiry;
  pet: Pet;
};

type ModalState = { inquiry: ?PetInquiry };

class PetModal extends React.Component {

  props: ModalProps;
  state: ModalState;
  onSubmitClick: () => void;

  constructor(props: ModalProps) {
    super(props);

    this.state = { inquiry: null };

    this.onSubmitClick = () => {
      props.sendInquiry({
        type: 'SEND_INQUIRY',
        action: props.pet.action || 'adopt',
        id: props.pet.id,
        inquiry: this.state.inquiry
      });
    };
  }

  render() { ... }
}
```

```
src/views/modal-example.jsx:25
25:           props.sendInquiry({
              ^ call of method 'sendInquiry'
29:             inquiry: this.state.inquiry
                           ^^^^^^^^^^^^^^ null. This type is incompatible with
57:             inquiry: PetInquiry;
                           ^^^^^^^^^^ object type. See lib: decls/shelter.js:57
```

```
src/views/modal-example.jsx:25
25:           props.sendInquiry({
              ^ call of method 'sendInquiry'
29:             inquiry: this.state.inquiry
                           ^^^^^^^^^^ undefined. This type is incompatible with
57:             inquiry: PetInquiry;
                           ^^^^^^^^^^ object type. See lib: decls/shelter.js:57
```

```
declare type PetInquiry = {
    name: string;
    email: string;
    zip: number;
}

declare type ActionSendInquiry = {
    type: 'SEND_INQUIRY';
    action: PetAction;
    id: number;
    inquiry: PetInquiry;
}

declare type SendInquiry =
(x: ActionSendInquiry) => void;

// @flow

type ModalProps = {
    sendInquiry: SendInquiry;
    pet: Pet;
};

type ModalState = { inquiry: ?PetInquiry };

class PetModal extends React.Component {

    props: ModalProps;
    state: ModalState;
    onSubmitClick: () => void;

    constructor(props: ModalProps) {
        super(props);

        this.state = { inquiry: null };

        this.onSubmitClick = () => {
            this.state.inquiry && props.sendInquiry({
                type: 'SEND_INQUIRY',
                action: props.pet.action || 'adopt',
                id: props.pet.id,
                inquiry: this.state.inquiry
            });
        };
    }

    render() {...}
}
```

No errors!

Thanks !

- <https://flowtype.org/>
- <https://github.com/coreygrunewald/pet-shelter>
- [Caml Trading - Yaron Minsky](#)
- <https://github.com/facebook/flow/blob/v0.32.0/lib/react.js>