

React Course

Session 3

Radoslav Stankov



**ASK MORE
QUESTIONS**

GET MORE ANSWERS
Anthony Burrill





Schedule

1. React basics (with TypeScript)
2. React advanced patterns

3. Forms

4. Redux basics
5. Redux advanced patterns
6. Advanced topics - Authorize / CSS in React / Router
7. Unit Testing
8. GraphQL / Apollo
9. Q&A

How not to hate your life



Dealing with forms in React

The Definitive Guide

O RLY?

Radoslav Stankov

ANMELDUNG bei der Meldebehörde

Einzugsdatum

Dienststelle

Neue Wohnung (Straße / Platz, Hausnummer, Stockwerk) (1)

Zustellpostamt

Berlin

Bei

(die Namensangabe des Wohnungsgebers ist freiwillig; sie wird empfohlen, wenn der Einwohner nicht durch eigenes Namensschild an Haus- und Wohnungstür erkennbar ist.)

Wird die bisherige Wohnung beibehalten? (3)
Hat eine der zur Anmeldung kommenden Personen eine weitere Wohnung? (4)
Wird die bisherige Wohnung nicht aufgegeben oder bestehen weitere Wohnungen, füllen S "Beiblatt zur Anmeldung bei mehreren Wohnungen" aus.

Bei Zuzug nach Berlin aus dem Ausland bitte hier gemäß Erläuterungen (5) gegebenenfalls die frühere Namen (z.B. Geburtsname, Ordens- oder Künstlernamen)

Lfd. Nr.	(6) Geburtsdatum	(7) Geschlecht (m., w.)	(8) Familienname, Doktorgrad
1			
2			
3			
4			

Lfd. Nr.	(10) Geburtsort (Gde., Lkr., falls Ausland: auch Staat angeben)	(11) Staatsangehörigkeit(en)	(12) Religion (nur ev., rk. oder ak.)
1			
2			
3			
4			

Lfd. Nr.	(15) (15) Identifikationsnummer nach § 139b der Abgabenordnung	(16) Lohnsteuerkarte für das folgende Jahr gewünscht	(17) Für Flüchtlinge/ Vertriebene Wohnsitz am 1. Sept. 1939 (Wohnt, Landkreis, Provinz)
1		<input type="checkbox"/> Ja <input type="checkbox"/> Nein	
2		<input type="checkbox"/> Ja <input type="checkbox"/> Nein	
3		<input type="checkbox"/> Ja <input type="checkbox"/> Nein	
4		<input type="checkbox"/> Ja <input type="checkbox"/> Nein	

Lfd. Nr.	(19) Personalausweis Seriennummer	Ausstellungs- behörde	Familienstand
1			
2			
3			



What makes working with forms hard in frontend?



KEEP
CALM
AND
TAKE A
STEP BACK

What makes working with forms hard in frontend?

- UI structure
- Dependent state
- Custom controls
- Error handling
- Integration with server



Agenda for Today

- Form basics
- React.Context
- Extract form components
- Formik
- Use Formik with form components

Form Basics

Submit a talk

Title:

How not to hate your life when dealing with forms

Email:

rado@example.com

Description:



Length:

30 minutes



Level:

- beginner
- intermediate
- expert

Speakers:

1 Radoslav Stankov remove

2 First Name Last Name remove

[add speaker](#)

Submit

Submit a talk

Title:

Email:

Description:

Length:

15 minutes



Level:

- beginner
- intermediate
- expert

Submit

Submit a talk

Title:

Submit

```
<form>
  <h1>
    <label>
      <input />
    </label>
    <input type="submit" />
</form>
```

Submit a talk

Title:

Submit

```
function SubmitTalkForm() {
  const [state, setState] = useState({});

  const onSubmit = event => {
    event.preventDefault();
    console.log(state);
 };

  return (
    <form onSubmit={onSubmit}>
      <div>
        <label htmlFor="title">Title:</label>
        <input
          id="title"
          type="text"
          name="title"
          value={state.title || ''}
          onChange={({ target }) =>
            setState({ ...state, title: target.value })
          }
        />
      </div>
      <input type="submit" value="Submit" />
    </form>
  );
}
```

```
function SubmitTalkForm() {
  const [state, setState] = useState({});

  const onSubmit = event => {
    event.preventDefault();
    console.log(state);
 };

  return (
    <form onSubmit={onSubmit}>
      <div>
        <label htmlFor="title">Title:</label>
        <input
          id="title"
          type="text"
          name="title"
          value={state.title || ''}
          onChange={({ target }) =>
            setState({ ...state, title: target.value })
          }
        />
      </div>
      <input type="submit" value="Submit" />
    </form>
  );
}
```

```
function SubmitTalkForm() {
  const [state, setState] = useState({});

  const onSubmit = event => {
    event.preventDefault();
    console.log(state);
 };

  return (
    <form onSubmit={onSubmit}>
      <div>
        <label htmlFor="title">Title:</label>
        <input
          id="title"
          type="text"
          name="title"
          value={state.title || ''}
          onChange={({ target }) =>
            setState({ ...state, title: target.value })
          }
        />
      </div>
      <input type="submit" value="Submit" />
    </form>
  );
}
```

Input change

```
function SubmitTalkForm() {
  const [state, setState] = useState({});

  const onSubmit = event => {
    event.preventDefault();
    console.log(state);
 };

  return (
    <form onSubmit={onSubmit}>
      <div>
        <label htmlFor="title">Title:</label>
        <input
          id="title"
          type="text"
          name="title"
          value={state.title || ''}
          onChange={({ target }) =>
            setState({ ...state, title: target.value })
          }
        />
      </div>
      <input type="submit" value="Submit" />
    </form>
  );
}
```

Input change



onChange

setState

```
function SubmitTalkForm() {
  const [state, setState] = useState({});

  const onSubmit = event => {
    event.preventDefault();
    console.log(state);
 };

  return (
    <form onSubmit={onSubmit}>
      <div>
        <label htmlFor="title">Title:</label>
        <input
          id="title"
          type="text"
          name="title"
          value={state.title || ''}
          onChange={({ target }) =>
            setState({ ...state, title: target.value })
          }
        />
      </div>
      <input type="submit" value="Submit" />
    </form>
  );
}
```

Input change

onChange

setState

render

New value

Submit a talk

Title:

Email:

Description:

Length:

15 minutes



Level:

- beginner
- intermediate
- expert

Submit

Submit a talk

Title:

Email:

Description:

Length:

15 minutes



Level:

- beginner
- intermediate
- expert

Submit

Field

Label

Input Control

Field

Label

?

Error

Input Control

Description

```
<form>
  <h1>
    <label>
      <input />
    </label>
    <input type="submit" />
</form>
```

Submit a talk

Title:

Submit

```
<form>  
  <h1>  
    <Field />  
  <input type="submit" />  
</form>
```

Submit a talk

Title:

Submit

```
function SubmitTalkForm() {
  const [state, setState] = useState({});

  const onSubmit = event => { /* */ };

  return (
    <form onSubmit={onSubmit}>
      <Field
        id="title"
        name="title"
        value={state.title || ''}
        onChange={({ target }) => setState({ ...state, title: target.value })}
      />
      <input type="submit" value="Submit" />
    </form>
  );
}
```

```
function Field({ label, name, ...inputProps }) {
  return (
    <div>
      <label htmlFor={inputProps.id}>{label || capitalize(name)}:</label>
      <input type="text" name={name} {...inputProps} />
    </div>
  );
}
```

```
function SubmitTalkForm() {
  const [state, setState] = useState({});

  const onSubmit = event => { /* */ };

  return (
    <form onSubmit={onSubmit}>
      <Field
        id="title"
        name="title"
        value={state.title || ''}
        onChange={({ target }) => setState({ ...state, title: target.value })}
      />
      <input type="submit" value="Submit" />
    </form>
  );
}
```

```
function Field({ label, name, ...inputProps }) {
  return (
    <div>
      <label htmlFor={inputProps.id}>{label || capitalize(name)}:</label>
      <input type="text" name={name} {...inputProps} />
    </div>
  );
}
```

React.Context

Form

Form

Some Component

Form

Some Component

Some Other Component

Form

Some Component

Some Other Component

Field

Form

Context

Some Component

Some Other Component

Field

Form

Context

Some Component

Some Other Component

Field

Context



React.Context History

React.Context

```
class App extends React.Component {
  render() {
    return <Toolbar theme="dark" />;
  }
}

function Toolbar(props) {
  return (
    <div>
      <ThemedButton theme={props.theme} />
    </div>
  );
}

class ThemedButton extends React.Component {
  render() {
    return <Button theme={this.props.theme} />;
  }
}
```

```
<html>
  <App>
    <Compnent1>
      <Compnent2>
        <Toolbar>
          <Component4>
            <Component5>
              <ThemedButton />
            </Component5>
          </Component4>
        </Toolbar>
      </Compnent2>
    </Compnent1>
  </ReactApp>
</App>
```

```
const ThemeContext = React.createContext('light');

function App() {
  return (
    <ThemeContext.Provider value="dark">
      <Toolbar />
    </ThemeContext.Provider>
  );
}

function Toolbar(props) {
  return (
    <div>
      <ThemedButton />
    </div>
  );
}

class ThemedButton extends React.Component {
  static contextType = ThemeContext;

  render() {
    return <Button theme={this.context} />;
  }
}
```

```
const ThemeContext = React.createContext('light');

function App() {
  return (
    <ThemeContext.Provider value="dark">
      <Toolbar />
    </ThemeContext.Provider>
  );
}

function Toolbar(props) {
  return (
    <div>
      <ThemedButton />
    </div>
  );
}

class ThemedButton extends React.Component {
  static contextType = ThemeContext;

  render() {
    return <Button theme={this.context} />;
  }
}
```

```
const ThemeContext = React.createContext('light');

function App() {
  return (
    <ThemeContext.Provider value="dark">
      <Toolbar />
    </ThemeContext.Provider>
  );
}

function Toolbar(props) {
  return (
    <div>
      <ThemedButton />
    </div>
  );
}

class ThemedButton extends React.Component {
  static contextType = ThemeContext;

  render() {
    return <Button theme={this.context} />;
  }
}
```

```
class ThemedButton extends React.Component {  
  static contextType = ThemeContext;  
  
  render() {  
    return <Button theme={this.context} />;  
  }  
}
```

```
function ThemedButton() {
  return (
    <ThemeContext.Consumer>
      {context => <Button theme={context} />}
    </ThemeContext.Consumer>
  );
}
```

```
function ThemedButton() {  
  const context = useContext(ThemeContext);  
  
  return (  
    <Button theme={context} />  
  );  
}
```

Submit a talk

Title:

Email:

Description:

Length:

15 minutes



Level:

- beginner
- intermediate
- expert

Submit

```
const FormContext = React.createContext();

function SubmitTalkForm() {
  const context = useState({});

  const onSubmit = event => { /* */ };

  return (
    <FormContext.Provider value={context}>
      <form onSubmit={onSubmit}>
        <Field name="title" />
        <input type="submit" value="submit" />
      </form>
    </FormContext.Provider>
  );
}

function Field({ label, name, ...inputProps }) {
  const [state, setState] = useContext(FormContext);

  return (
    <div>
      <label htmlFor={name}>{label || capitalize(name)}: </label>
      <input
        type="text"
        id={name}
        name={name}
        value={state[name] || ''}
      </input>
    </div>
  );
}
```

```
const FormContext = React.createContext();

function SubmitTalkForm() {
  const context = useState({});

  const onSubmit = event => { /* */ };

  return (
    <FormContext.Provider value={context}>
      <form onSubmit={onSubmit}>
        <Field name="title" />
        <input type="submit" value="submit" />
      </form>
    </FormContext.Provider>
  );
}

function Field({ label, name, ...inputProps }) {
  const [state, setState] = useContext(FormContext);

  return (
    <div>
      <label htmlFor={name}>{label || capitalize(name)}: </label>
      <input
        type="text"
        id={name}
        name={name}
        value={state[name] || ''}>
    
```

```
const FormContext = React.createContext();

function SubmitTalkForm() {
  const context = useState({});

  const onSubmit = event => { /* */ };

  return (
    <FormContext.Provider value={context}>
      <form onSubmit={onSubmit}>
        <Field name="title" />
        <input type="submit" value="submit" />
      </form>
    </FormContext.Provider>
  );
}

function Field({ label, name, ...inputProps }) {
  const [state, setState] = useContext(FormContext);

  return (
    <div>
      <label htmlFor={name}>{label || capitalize(name)}: </label>
      <input
        type="text"
        id={name}
        name={name}
        value={state[name] || ''}
      </input>
    </div>
  );
}
```

```
return (
  <FormContext.Provider value={context}>
    <form onSubmit={onSubmit}>
      <Field name="title" />
      <input type="submit" value="submit" />
    </form>
  </FormContext.Provider>
);
}

function Field({ label, name, ...inputProps }) {
  const [state, setState] = useContext(FormContext);

  return (
    <div>
      <label htmlFor={name}>{label || capitalize(name)}: </label>
      <input
        type="text"
        id={name}
        name={name}
        value={state[name] || ''}
        onChange={({ target }) => setState({ ...state, [name]: target.value })}
        {...inputProps}
      />
    </div>
  );
}
```

```
<form>  
  <h1>  
    <Field />  
  <input type="submit" />  
</form>
```

Submit a talk

Title:

Submit

```
<Form>
  <h1>
    <Form.Field />
  <Form.Submit />
</Form>
```

Submit a talk

Title:

Submit

```
function SubmitTalkForm() {
  const onSubmit = values => {
    console.log(value);
  };

  return (
    <Form onSubmit={onSubmit}>
      <Form.Field name="title" />
      <Form.Submit />
    </Form>
  );
}
```

```
const FormContext = React.createContext([{}, function() {}]);\n\nfunction Form({ onSubmit, initialValues, ...props }) {\n  const context = useState(initialValues);\n\n  function handleSubmit(e) {\n    e.preventDefault();\n    onSubmit(context[0]);\n  }\n\n  return (\n    <FormContext.Provider value={context}>\n      <form onSubmit={handleSubmit} {...props} />\n    </FormContext.Provider>\n  );\n}\n\nForm.Field = Field;\n\nForm.Submit = () => <input type="submit" value="submit" />;
```

```
<Form>
  <h1>
    <Form.Field />
  <Form.Submit />
</Form>
```

Submit a talk

Title:

Submit

```
<Form>
  <h1>
    <Form.Field />
    <Form.Field />
    <Form.Submit />
</Form>
```

Submit a talk

Title:

Email:

Submit

```
<Form>
  <Form.Field name="title" />
  <Form.Field name="email" control="email" />
  <Form.Submit />
</Form>
```

```
function Field({ label, name, control, ...inputProps }) {  
  const [state, setState] = useContext(FormContext);  
  
  return (  
    <div>  
      <label htmlFor={name}>{label || capitalize(name)}:</label>  
      <input  
        type="text"  
        id={name}  
        name={name}  
        value={state[name] || ''}  
        onChange={({ target }) => setState({ ...state, [name]: target.value }) }  
        {...inputProps} />  
    </div>  
  );  
}
```

```
function Field({ label, name, control, ...inputProps }) {  
  const [state, setState] = useContext(FormContext);  
  
  return (  
    <div>  
      <label htmlFor={name}>{label || capitalize(name)}:</label>  
      <input  
        type={control || 'text'}  
        id={name}  
        name={name}  
        value={state[name] || ''}  
        onChange={({ target }) => setState({ ...state, [name]: target.value }) }  
        {...inputProps} />  
    </div>  
  );  
}
```

```
<Form>
  <h1>
    <Form.Field />
    <Form.Field />
    <Form.Submit />
</Form>
```

Submit a talk

Title:

Email:

Submit

```
<Form>
  <h1>
    <Form.Field />
    <Form.Field />
    <Form.Field />
    <Form.Submit />
</Form>
```

Submit a talk

Title:

Email:

Description:

Submit

```
<Form>
  <Form.Field name="title" />
  <Form.Field name="email" control="email" />
  <Form.Field name="description" control="textarea" />
  <Form.Submit />
</Form>
```

```
function Field({ label, name, control, ...inputProps }) {  
  const [state, setState] = useContext(FormContext);  
  
  return (  
    <div>  
      <label htmlFor={name}>{label || capitalize(name)}</label>  
      <input  
        type={control || 'text'}  
        id={name}  
        name={name}  
        value={state[name] || ''}  
        onChange={({ target }) => setState({ ...state, [name]: target.value }) }  
        {...inputProps} />  
    </div>  
  );  
}
```

```
function Field({ label, name, control, ...inputProps }) {
  const [state, setState] = useContext(FormContext);

  let Control = props => <input type={control || 'text'} {...props} />;

  if (control === 'textarea') {
    Control = props => <textarea {...props} />;
  }

  return (
    <div>
      <label htmlFor={name}>{label || capitalize(name)}:</label>
      <Control
        id={name}
        name={name}
        value={state[name] || ''}
        onChange={({ target }) => setState({ ...state, [name]: target.value })}
        {...inputProps}
      />
    </div>
  );
}
```

```
function Field({ label, name, control, ...inputProps }) {
  const [state, setState] = useContext(FormContext);

  let Control = props => <input type={control || 'text'} {...props} />;

  if (control === 'textarea') {
    Control = props => <textarea {...props} />;
  }

  return (
    <div>
      <label htmlFor={name}>{label || capitalize(name)}:</label>
      <Control
        id={name}
        name={name}
        value={state[name] || ''}
        onChange={({ target }) => setState({ ...state, [name]: target.value })}
        {...inputProps}
      />
    </div>
  );
}
```

```
<Form>
  <h1>
    <Form.Field />
    <Form.Field />
    <Form.Field />
    <Form.Submit />
</Form>
```

Submit a talk

Title:

Email:

Description:

Submit

```
<Form>
  <h1>
    <Form.Field />
    <Form.Field />
    <Form.Field />
    <Form.Field />
    <Form.Submit />
</Form>
```

Submit a talk

Title:

Email:

Description:

Length:

▼

Submit

```
const LENGTH_OPTIONS = [
  { value: 15, label: '15 minutes' },
  { value: 30, label: '30 minutes' },
  { value: 45, label: '45 minutes' },
];
<Form>
  <Form.Field name="title" />
  <Form.Field name="email" control="email" />
  <Form.Field name="description" control="textarea" />
  <Form.Field name="length" control="select" options={LENGTH_OPTIONS} />
  <Form.Submit />
</Form>
```

```
function Field({ label, name, control, ...inputProps }) {
  const [state, setState] = useContext(FormContext);

  let Control = props => <input type={control || 'text'} {...props} />;

  if (control === 'textarea') {
    Control = props => <textarea {...props} />;
  }

  if (control === 'select') {
    Control = ({ options, ...props }) => (
      <select {...props}>
        {options.map(({ label, value }, i) => (
          <option key={i} value={value}>
            {label || value}
          </option>
        ))}
      </select>
    );
  }

  return (
    <div>
      <label htmlFor={name}>{label || capitalize(name)}:</label>
      <Control
        id={name}
        name={name}
      >
    
```

```
function Field({ label, name, control, ...inputProps }) {
  const [state, setState] = useContext(FormContext);

  let Control = props => <input type={control || 'text'} {...props} />;

  if (control === 'textarea') {
    Control = props => <textarea {...props} />;
  }

  if (control === 'select') {
    Control = ({ options, ...props }) => (
      <select {...props}>
        {options.map(({ label, value }, i) => (
          <option key={i} value={value}>
            {label || value}
          </option>
        ))}
      </select>
    );
  }

  return (
    <div>
      <label htmlFor={name}>{label || capitalize(name)}:</label>
      <Control
        id={name}
        name={name}
      >
    
```

```
const CONTROLS = {
  undefined: props => <input type="text" {...props} />,
  text: props => <input type="text" {...props} />,
  email: props => <input type="email" {...props} />,
  textarea: props => <textarea {...props} />,
  select: ({ options, ...props }) => (
    <select {...props}>
      {options.map(({ label, value }, i) => (
        <option key={i} value={value}>
          {label || value}
        </option>
      ))}
    </select>
  ),
};

function Field({ label, name, control, ...inputProps }) {
  const [state, setState] = useContext(FormContext);

  const Control = CONTROLS[control];

  return (
    <div>
      <label htmlFor={name}>{label || capitalize(name)}:</label>
      <Control
        id={name}
        name={name}
        state={state}
        setState={setState}
        {...inputProps}
      />
    </div>
  );
}
```

```
const CONTROLS = {
  undefined: props => <input type="text" {...props} />,
  text: props => <input type="text" {...props} />,
  email: props => <input type="email" {...props} />,
  textarea: props => <textarea {...props} />,
  select: ({ options, ...props }) => (
    <select {...props}>
      {options.map(({ label, value }, i) => (
        <option key={i} value={value}>
          {label || value}
        </option>
      ))}
    </select>
  ),
};

function Field({ label, name, control, ...inputProps }) {
  const [state, setState] = useContext(FormContext);

  const Control = CONTROLS[name];

  return (
    <div>
      <label htmlFor={name}>{label || capitalize(name)}:</label>
      <Control
        id={name}
        name={name}
        state={state}
        setState={setState}
        {...inputProps}
      />
    </div>
  );
}
```

```
<Form>
  <h1>
    <Form.Field />
    <Form.Field />
    <Form.Field />
    <Form.Field />
    <Form.Submit />
</Form>
```

Submit a talk

Title:

Email:

Description:

Length:

▼

Submit

```
<Form>
  <h1>
    <Form.Field />
    <Form.Field />
    <Form.Field />
    <Form.Field />
    <Form.Field />
    <Form.Submit />
</Form>
```

Submit a talk

Title:

Email:

Description:

Length:

 ▼

Level:

- beginner
- intermediate
- expert

Submit

```
const CONTROLS = {
  /* */
  radioGroup: ({ value, options, name, onChange, ...props }) => (
    <ul>
      {options.map((option, value), i) => (
        <li key={i}>
          <label>
            <input
              type="radio"
              name={name}
              value={option.value}
              checked={option.value === value}
              onChange={onChange}
              {...props}
            />
            {option.label || option.value}
          </label>
        </li>
      ))}
    </ul>
  ),
};
```

```
<Form>
  <h1>
    <Form.Field />
    <Form.Field />
    <Form.Field />
    <Form.Field />
    <Form.Field />
    <Form.Submit />
</Form>
```

Submit a talk

Title:

Email:

Description:

Length:

 ▼

Level:

- beginner
- intermediate
- expert

Submit

```
<Form>
  <h1>
    <Form.Field />
    <Form.Field />
    <Form.Field />
    <Form.Field />
    <Form.Field />
    <Form.Field />
    <Form.Submit />
</Form>
```

Submit a talk

Title:

Email:

Description:

Length:

▼

Level:

beginner

intermediate

expert

Speakers:

1 remove

[add speaker](#)

```
<Form>
  <Form.Field name="title" />
  <Form.Field name="email" control="email" />
  <Form.Field name="description" control="textarea" />
  <Form.Field name="length" control="select" options={LENGTH_OPTIONS} />
  <Form.Field name="level" control="radioGroup" options={LEVEL_OPTIONS} />
  <Form.Field name="speakers" control={SpeakersInput} />
  <Form.Submit />
</Form>
```

```
function Field({ label, name, control, ...inputProps }) {
  const [state, setState] = useContext(FormContext);

  const Control = CONTROLS[control];

  return (
    <div>
      <label htmlFor={name}>{label || capitalize(name)}:</label>
      <Control
        id={name}
        name={name}
        value={state[name] || ''}
        onChange={({ target }) => setState({ ...state, [name]: target.value })}
        {...inputProps}
      />
    </div>
  );
}
```

```
function Field({ label, name, control, ...inputProps }) {
  const [state, setState] = useContext(FormContext);

  const Control = typeof control === 'function' ? control : CONTROLS[control];

  return (
    <div>
      <label htmlFor={name}>{label || capitalize(name)}</label>
      <Control
        id={name}
        name={name}
        value={state[name] || ''}
        onChange={({ target }) => setState({ ...state, [name]: target.value })}
        {...inputProps}
      />
    </div>
  );
}
```

```
function Field({ label, name, control, ...inputProps }) {
  const [state, setState] = useContext(FormContext);

  const Control = typeof control === 'function' ? control : CONTROLS[control];

  return (
    <div>
      <label htmlFor={name}>{label || capitalize(name)}:</label>
      <Control
        id={name}
        name={name}
        value={state[name] || ''}
        onChange={({ target }) => setState({ ...state, [name]: target.value })}
        {...inputProps}
      />
    </div>
  );
}
```

Submit a talk

Title:

Email:

Description:

Length:

15 minutes



Level:

- beginner
- intermediate
- expert

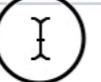
Speakers:

[add speaker](#)

[Submit](#)

Submit a talk

Title:

Email:

Description:

Length:

15 minutes



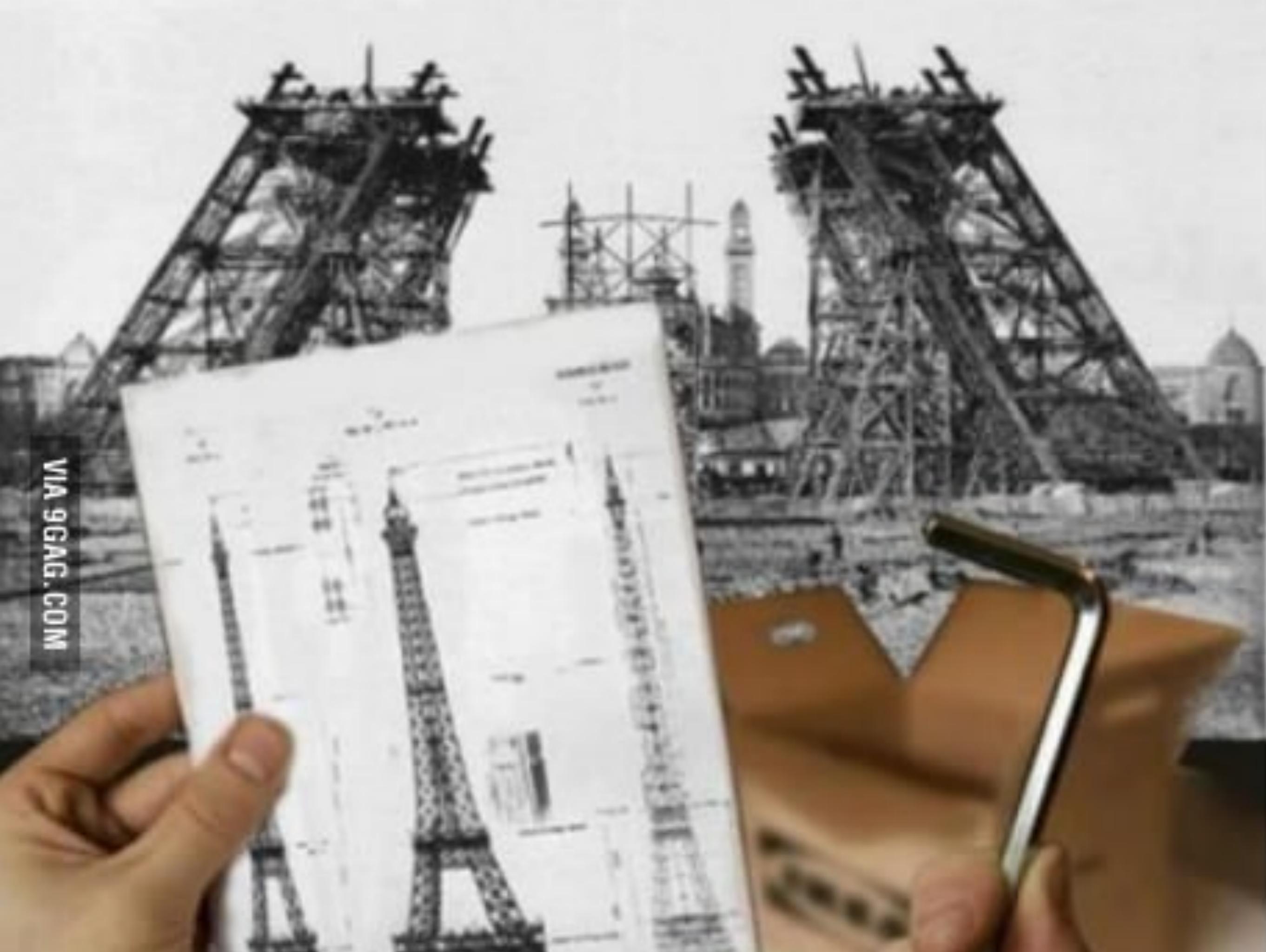
Level:

- beginner
- intermediate
- expert

Speakers:

[add speaker](#)

[Submit](#)



VIA GAG.COM



So far...

- simple form interface
- extensible fields
- standardized form layout



What is missing?

- validations - client / server side
- protection again double submit
- nested fields
- handle submit
- ... more

Libraries



Good Choice

Bad Choice



Libraries

- 🚚 <https://github.com/tannerlinsley/react-form>
- 🚌 <https://github.com/prometheusresearch/react-forms>
- 🚕 <https://github.com/codecks-io/react-reform>
- 🚗 <https://github.com/erikras/redux-form>
- 🚙 <https://github.com/Semantic-Org/Semantic-UI-React>
- 🚓 <https://github.com/jaredpalmer/formik>
- 🏍 <https://github.com/final-form/final-form>

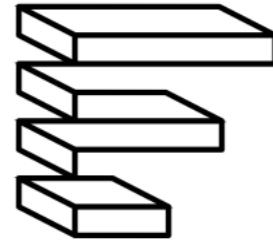


Final Form

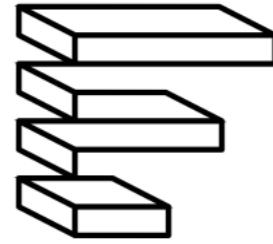


Formik



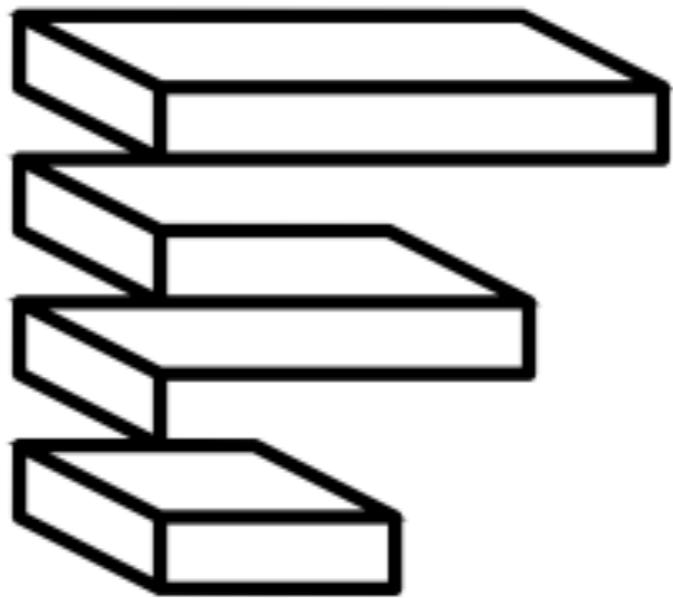


Formik



Formik





Formik

<https://jaredpalmer.com/formik/>

```
export default function Page() {
  return (
    <Formik initialValues={initialValues} validate={validate} onSubmit={submit}>
      {form => (
        <form onSubmit={form.handleSubmit}>
          <ErrorMessage name="title" />
          <Field id="title" name="title" className="form-control" />
          <input type="submit" disabled={form.isSubmitting} />
        </form>
      )}
    </Formik>
  );
}
```

```
export default function Page() {
  return (
    <Formik initialValues={initialValues} validate={validate} onSubmit={submit}>
      {form =>
        <form onSubmit={form.handleSubmit}>
          <div className="row justify-content-md-center">
            <div className="col-6">
              <div className="card">
                <div className="card-body">
                  <h1 className="card-title">Submit a talk</h1>

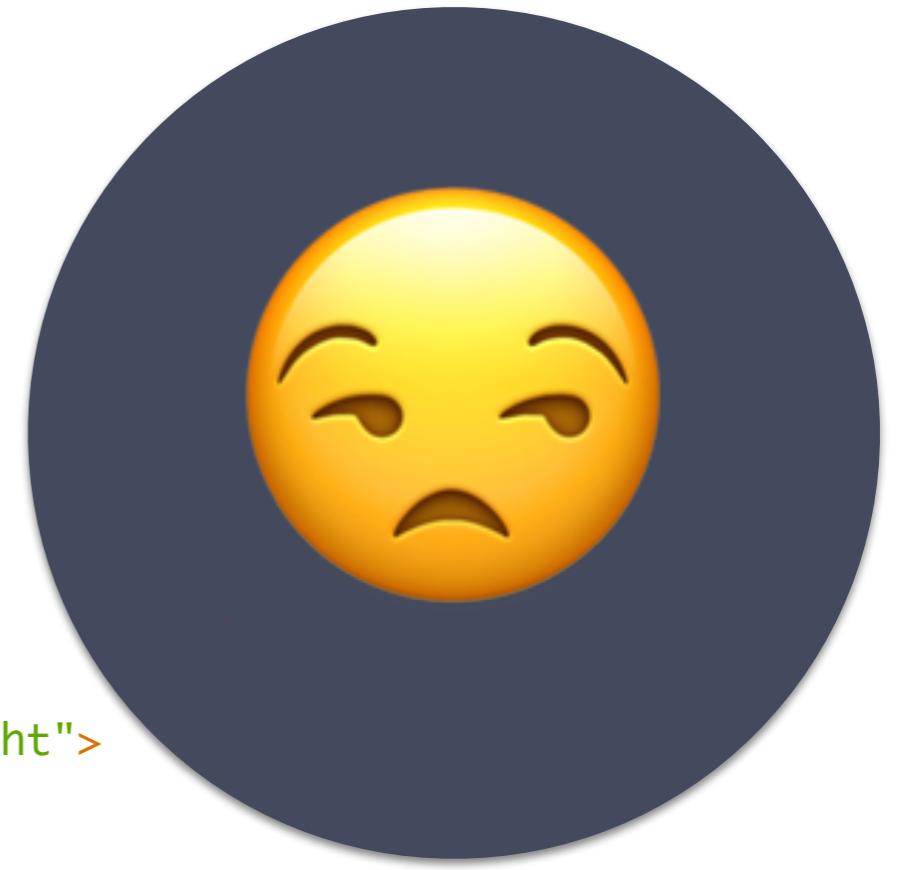
                  <div className="form-group d-block">
                    <div className="form-label d-block">
                      <label htmlFor="title">Title:</label>
                      <ErrorMessage name="title">
                        {error => (
                          <span className="text-danger float-right">
                            {error}
                          </span>
                        )}
                      </ErrorMessage>
                    </div>
                    <Field id="title" name="title" className="form-control" />
                  </div>

                  <div className="form-group d-block">
                    <div className="form-label d-block">
                      <label htmlFor="email">Email:</label>
                      <ErrorMessage name="email">
                        {error => (
                          <span className="text-danger float-right">
                            {error}
                          </span>
                        )}
                      </ErrorMessage>
                    </div>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </form>
      }
    )
  )
}
```

```
export default function Page() {
  return (
    <Formik initialValues={initialValues} validate={validate} onSubmit={submit}>
      {form =>
        <form onSubmit={form.handleSubmit}>
          <div className="row justify-content-md-center">
            <div className="col-6">
              <div className="card">
                <div className="card-body">
                  <h1 className="card-title">Submit a talk</h1>

                  <div className="form-group d-block">
                    <div className="form-label d-block">
                      <label htmlFor="title">Title:</label>
                      <ErrorMessage name="title">
                        {error => (
                          <span className="text-danger float-right">
                            {error}
                          </span>
                        )}
                      </ErrorMessage>
                    </div>
                    <Field id="title" name="title" className="form-control" />
                  </div>

                  <div className="form-group d-block">
                    <div className="form-label d-block">
                      <label htmlFor="email">Email:</label>
                      <ErrorMessage name="email">
                        {error => (
                          <span className="text-danger float-right">
                            {error}
                          </span>
                        )}
                      </ErrorMessage>
                    </div>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </form>
      }
    </Formik>
  )
}
```



```
<Form>
  <Form.Field name="title" />
  <Form.Field name="email" control="email" />
  <Form.Field name="description" control="textarea" />
  <Form.Field name="length" control="select" options={LENGTH_OPTIONS} />
  <Form.Field name="level" control="radioGroup" options={LEVEL_OPTIONS} />
  <Form.Field name="speakers" control={SpeakersInput} />
  <Form.Submit />
</Form>
```

<Form>

```
<Form.Field name="title" />
<Form.Field name="email" control="email" />
<Form.Field name="description" control="textarea" />
<Form.Field name="length" control="select" options={LENGTH_OPTIONS} />
<Form.Field name="level" control="radioGroup" options={LEVEL_OPTIONS} />
<Form.Field name="speakers" control={SpeakersInput} />
<Form.Submit />
</Form>
```

Form



```
<Form>
  <Form.Field name="title" />
  <Form.Field name="email" control="email" />
  <Form.Field name="description" control="textarea" />
  <Form.Field name="length" control="select" options={LENGTH_OPTIONS} />
  <Form.Field name="level" control="radioGroup" options={LEVEL_OPTIONS} />
  <Form.Field name="speakers" control={SpeakersInput} />
  <Form.Submit />
</Form>
```

FORMIK WRAPPER



Talk Is Cheap
Show Me the Code

ORLY?

Radoslav Stankov

```
{  
  "values": {  
    "title": "a",  
    "email": "",  
    "description": "",  
    "length": "",  
    "level": "",  
    "speakers": []  
  },  
  "errors": {  
    "description": "required",  
    "level": "required"  
  },  

```

<Form.State>
 {form => console.log(form)}
</Form.State>



```
<Form.State>
  {form => console.log(form)}
</Form.State>
```

```
{
  "values": {
    "title": "a",
    "email": "",
    "description": "",
    "length": "",
    "level": "",
    "speakers": []
  },
  "errors": {
    "description": "required",
    "level": "required"
  },
  "touched": {
    "title": true
  },
  "isSubmitting": false,
  "isValidating": false,
  "submitCount": 0,
  "dirty": true,
  "isValid": false,
  "initialValues": {
    "title": "",
    "email": "",
    "description": "",
    "length": "",
    "level": "",
    "speakers": []
  },
  "validateOnChange": true,
  "validateOnBlur": true
}
```

Submit a talk

Title:

Email:

Description:

Length:

 ▼

Level:

- beginner
- intermediate
- expert

Speakers:

1 remove

2 remove

3 remove

[add speaker](#)

Submit

15 minutes

Level:

- beginner
- intermediate
- expert

Speakers:

1	First Name	Last Name	remove
2	First Name	Last Name	remove
3	First Name	Last Name	remove

[add speaker](#)

Submit

```
<FieldArray name="speakers">
  {arrayHelpers =>
    <React.Fragment>
      {form.values.speakers.map(
        (_value: any, index: number) => (
          <Field name={`${arrayHelpers.name}.${index}.firstName`} />
          <Field name={`${arrayHelpers.name}.${index}.lastName`} />
          <button onClick={() => { arrayHelpers.remove(index); }}>remove</button>
        ),
      )}
      <button onClick={() => { arrayHelpers.push({ firstName: '', lastName: '' }, {}) }>add</button>
    </React.Fragment>
  }
</FieldArray>
```

```
<FieldArray name="speakers">
  {arrayHelpers =>
    <React.Fragment>
      {form.values.speakers.map(
        (_value: any, index: number) => (
          <Field name={`${arrayHelpers.name}.${index}.firstName`} />
          <Field name={`${arrayHelpers.name}.${index}.lastName`} />
          <button onClick={() => { arrayHelpers.remove(index); }}>remove</button>
        ),
      )}
      <button onClick={() => { arrayHelpers.push({ firstName: '', lastName: '' }, {}) }>add</button>
    </React.Fragment>
  }
</FieldArray>
```

Submit a talk

Title:

How not to hate your life when dealing with forms

Email:

rado@example.com

Description:



Length:

30 minutes



Level:

- beginner
- intermediate
- expert

Speakers:

1 Radoslav Stankov remove

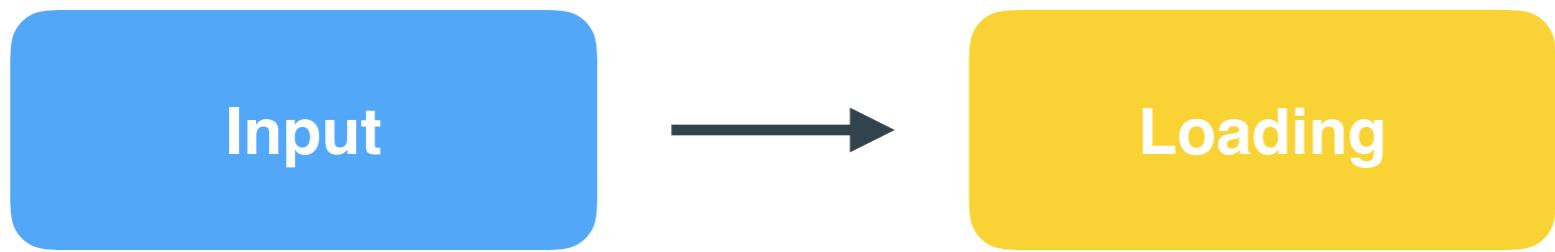
2 First Name Last Name remove

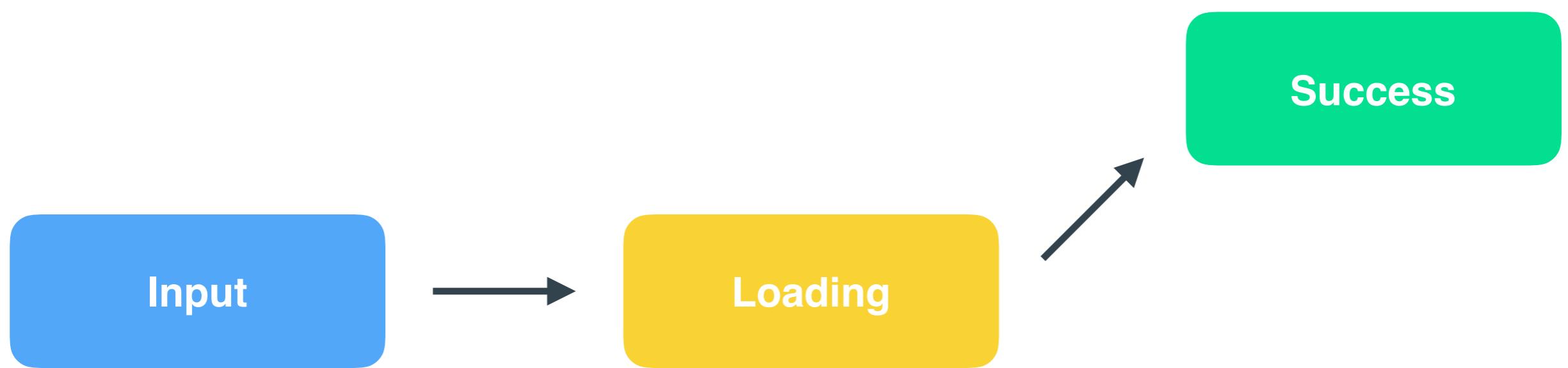
[add speaker](#)

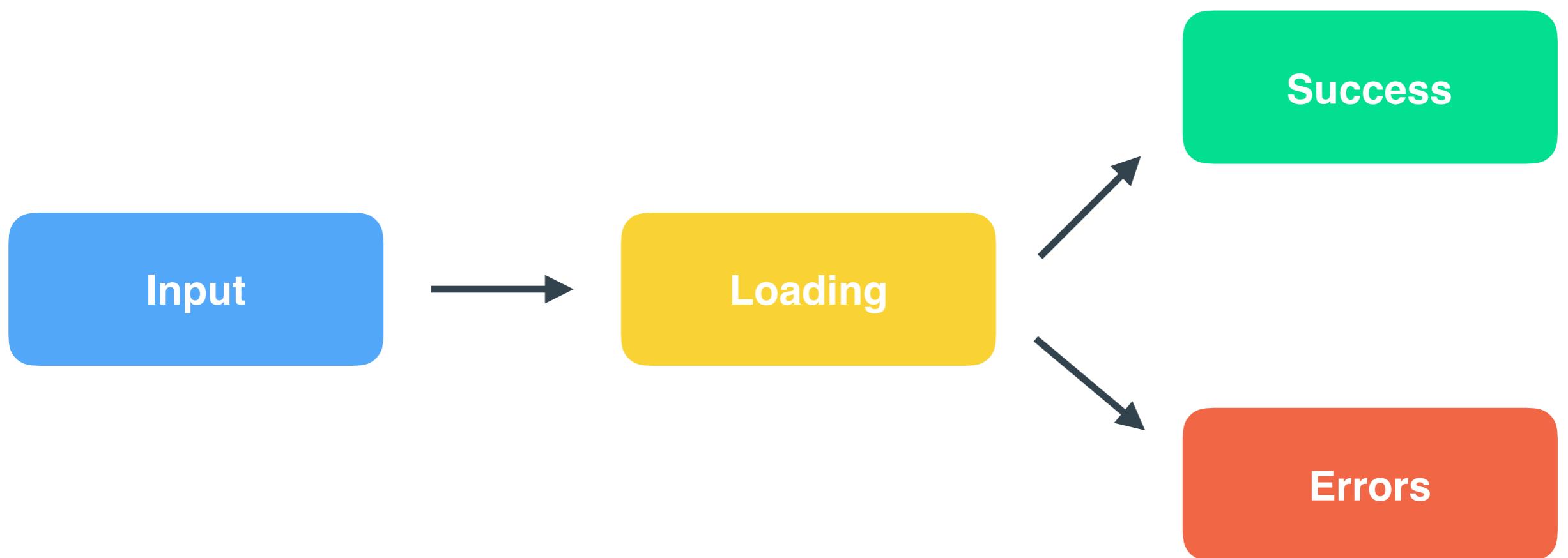
Submit

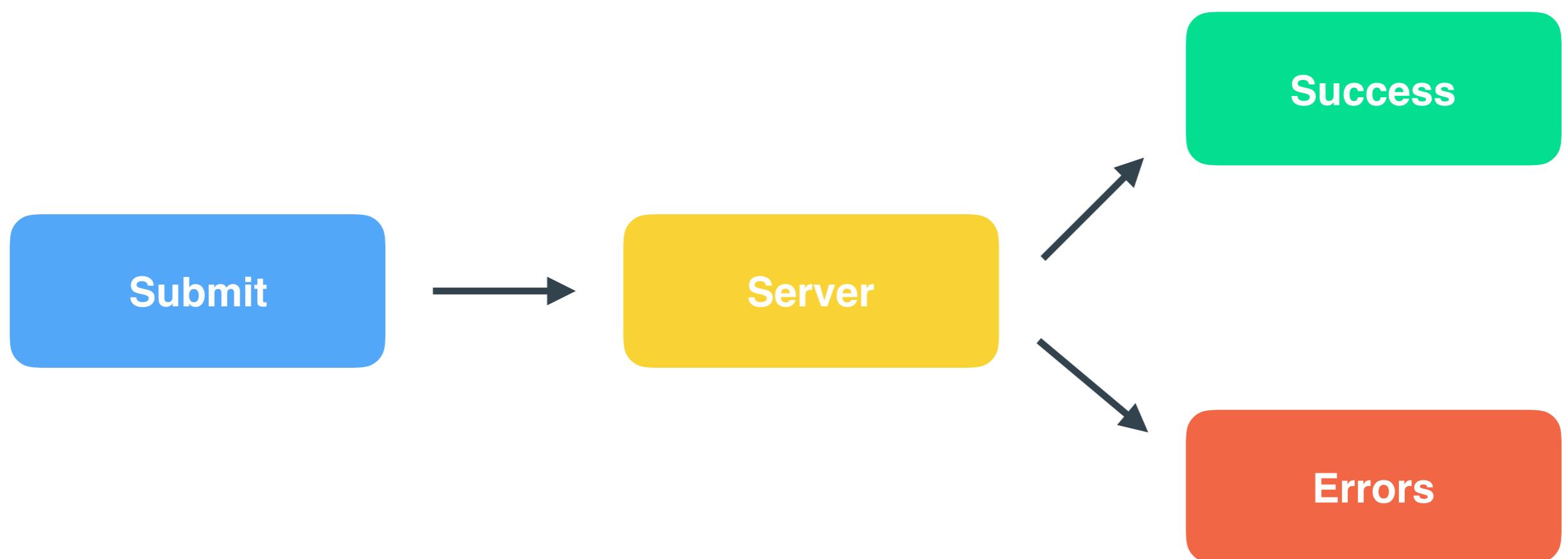
Server integration

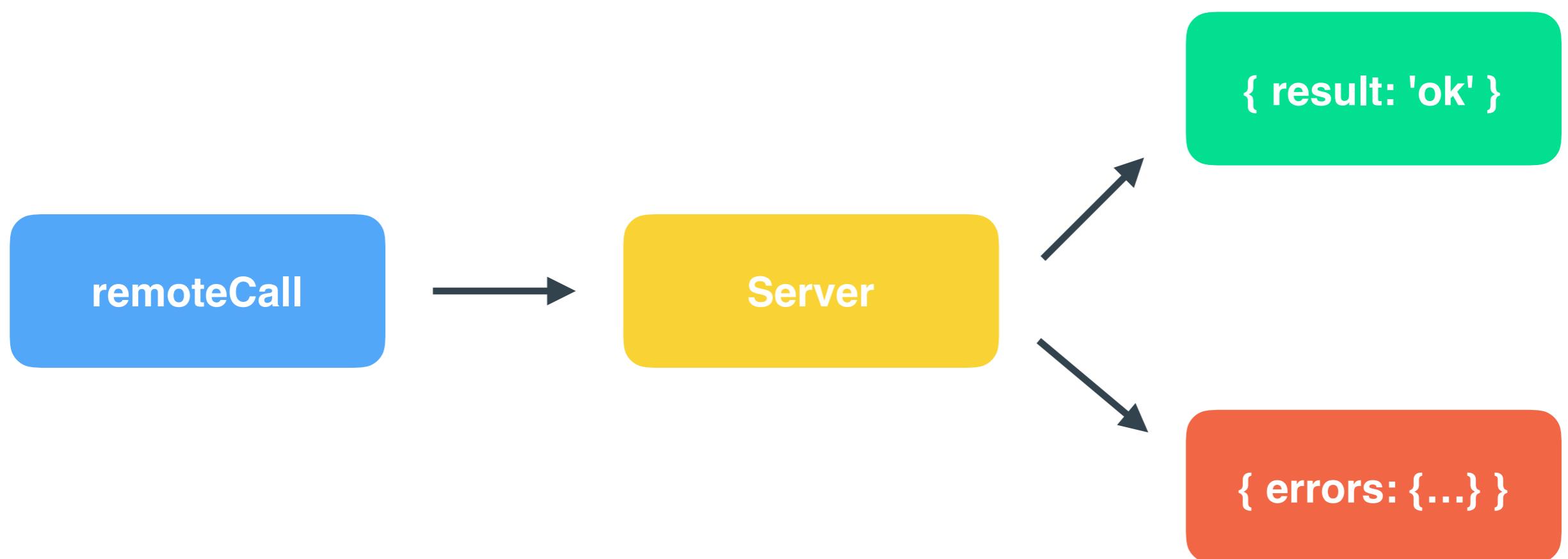
Input

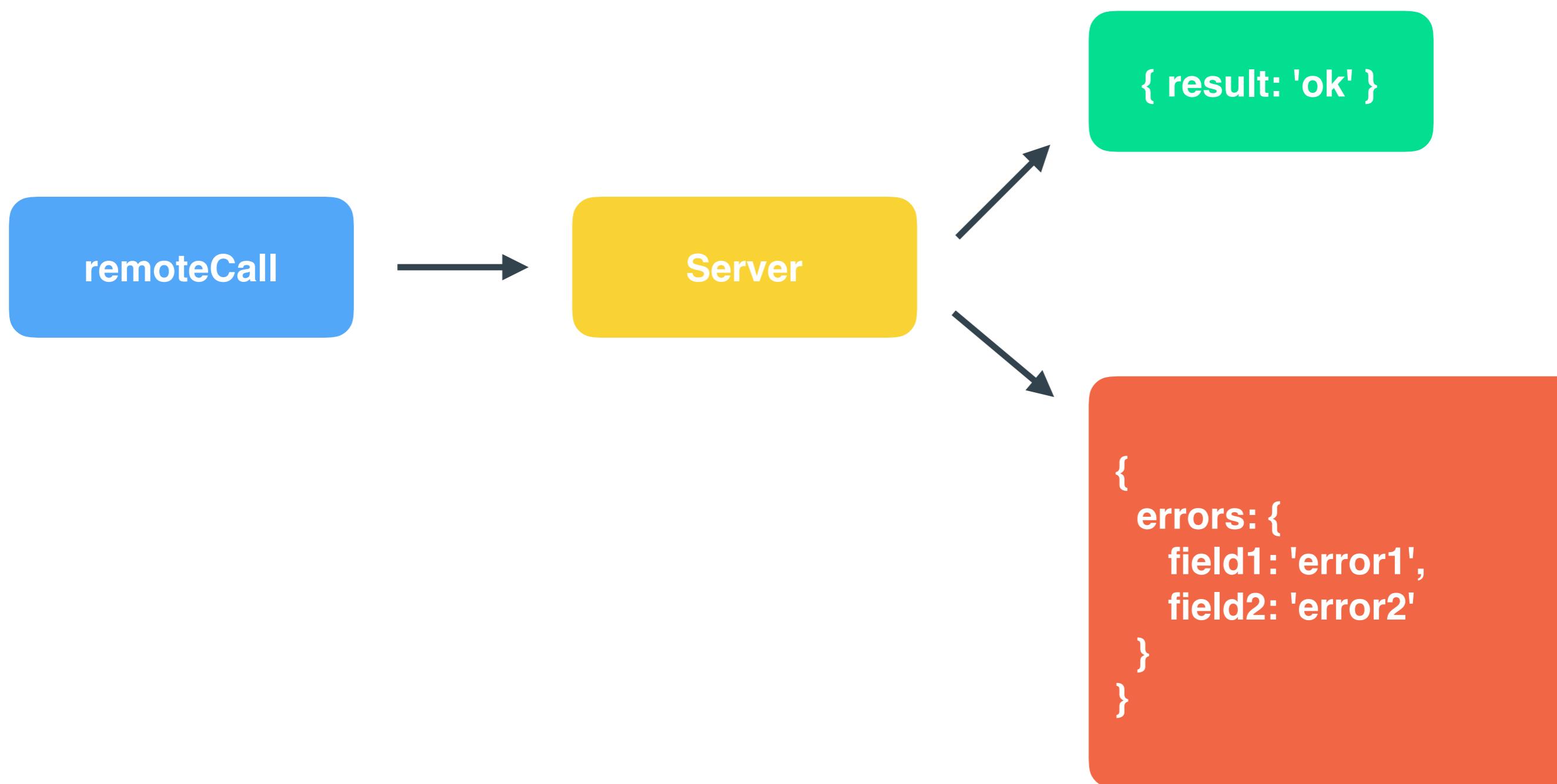












```
{  
  errors: {  
    field1: 'error1',  
    field2: 'error2'  
  }  
}
```

```
{  
  errors: {  
    field1: 'error1',  
    field2: 'error2'  
  }  
}
```

```
function submit(values, { setSubmitting, setErrors }) {
  // send values to server
  // we call setSubmitting(false)
  // if there are errors we call setErrors
}
```

Homework

Group 2 – Due in 7th December

Practise these at speed every day until you are really fast

1	$1 \times 9 = 9$	7	$7 \times 9 = 63$	13	$9 + 9 = 18$	19	$63 \div 9 = 7$
2	$2 \times 9 = 18$	8	$8 \times 9 = 72$	14	$18 + 9 = 27$	20	$72 \div 9 = 8$
3	$3 \times 9 = 27$	9	$9 \times 9 = 81$	15	$27 + 9 = 36$	21	$81 \div 9 = 9$
4	$4 \times 9 = 36$	10	$10 \times 9 = 90$	16	$36 + 9 = 45$	22	$90 \div 9 = 10$
5	$5 \times 9 = 45$	11	$11 \times 9 = 99$	17	$45 + 9 = 54$	23	$99 \div 9 = 11$
6	$6 \times 9 = 54$	12	$12 \times 9 = 108$	18	$54 + 9 = 63$	24	$108 \div 9 = 12$

Well done! I do 100 points.

Group 2 – due 21st June 2019

Practise these at speed every day until you are really fast

1	$1 \times 9 =$	7	$7 \times 9 =$	13	$1 - 9 =$	19	$63 \div 9 =$
2	$2 \times 9 =$	8	$8 \times 9 =$	14	$18 - 9 =$	20	$72 \div 9 =$
3	$3 \times 9 =$	9	$9 \times 9 =$	15	$27 - 9 =$	21	$81 \div 9 =$
4	$4 \times 9 =$	10	$10 \times 9 =$	16	$36 - 9 =$	22	$90 \div 9 =$
5	$5 \times 9 =$	11	$11 \times 9 =$	17	$45 - 9 =$	23	$99 \div 9 =$
6	$6 \times 9 =$	12	$12 \times 9 =$	18	$54 - 9 =$	24	$108 \div 9 =$

Complete the calculations

Division facts

Circle the smallest fraction

$\frac{3}{7}$ $\frac{1}{2}$ $\frac{1}{3}$

7 1 4 3
3 8 9
2 1

Shopping Cart

Pizzas

	Count	Price	
Pizza 1	2	\$20	remove
Pizza 2	2	\$30	remove
Pizza 3	2	\$40	remove
Pizza 4	2	\$50	remove

Total: \$140

Contact

Name*

Phone*

E-mail*

Address*

Submit

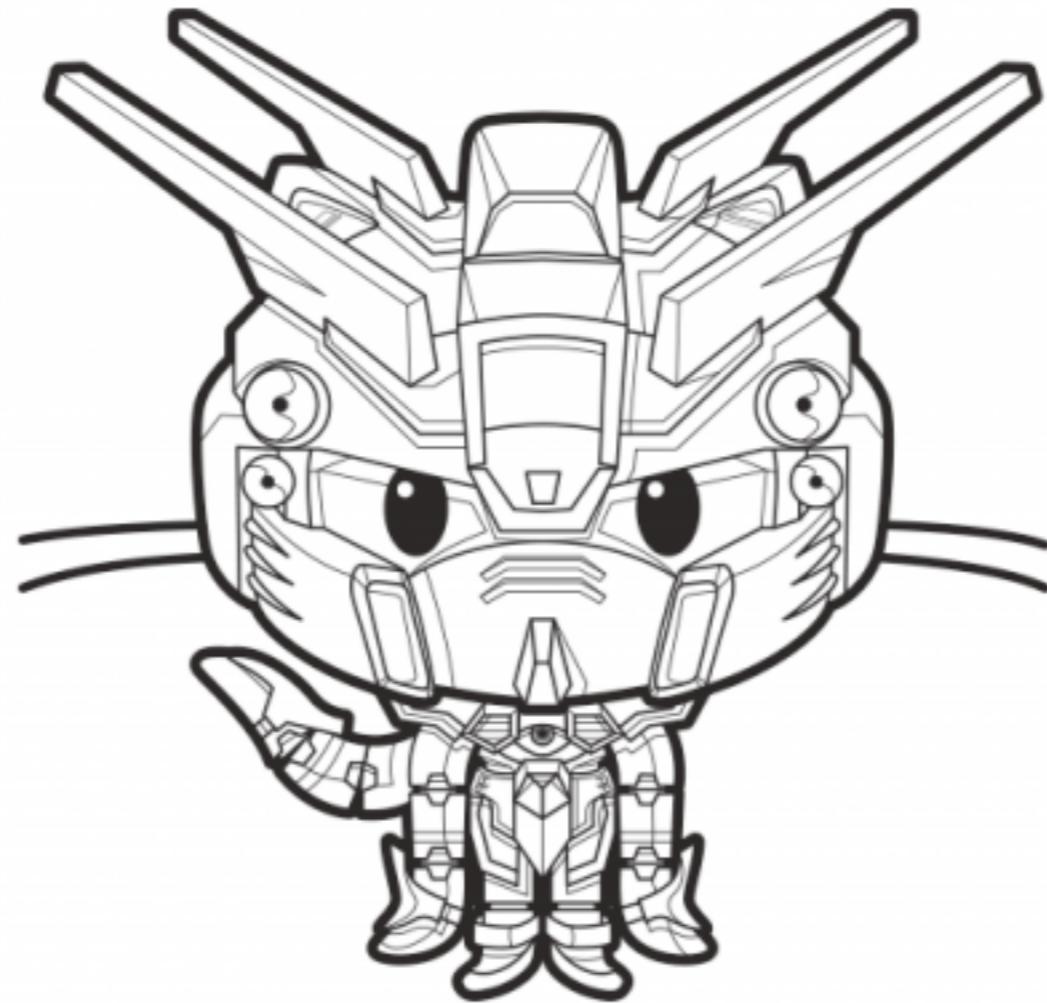
Homework #3

- Create shopping order form
 - Validate form information
 - Log form result in console
 - Create two versions
 - Version 1: use Formik directly
 - Version 2: use our Form component

**ASK MORE
QUESTIONS**

GET MORE ANSWERS
Anthony Burrill





<https://github.com/RStankov/react-course-mentor-mate>

