

Form validation with Formik and Folktale

Agenda

1. Formik
2. Folktale
3. module folktale/validation
4. Example

Formik

Handles the three following tasks for you:

1. Getting values in and out of form state
2. Validation and error messages
3. Handling form submission

Formik

```
import { Formik } from 'formik';
import { LoginForm } from './LoginForm'

export function Login() {
  return (
    <Formik
      component={LoginForm}
      initialValues={{ userName: '', password: '' }}
      onSubmit={handleSubmit}
    />
  )
}
```

Form and Fields

```
import { Field, Form } from 'formik';

export function LoginForm() {
  return (
    <Form>
      <Field
        type="text"
        name="userName"
        placeholder="Enter user name"
      />
    </Form>
  )
}
```

From and Fields

```
import { Field, Form } from 'formik';
import { PasswordInput } from './PasswordInput';

export function LoginForm() {
  return (
    <Form>
      <Field
        name="password"
        component={PasswordInput}
      />
    </Form>
  )
}
```

Form Submission

```
function handleSubmit(formValues, formikActions) {  
  // ...  
  
  formikActions.setSubmitting(false)  
}
```


Validation

```
function validateForm(formValues) {  
  //...  
  
  return {  
    userName: "Name can't be empty.",  
    password: "Password can't be empty.",  
  }  
}
```

```
<Formik validate={validateForm} />
```

Folktale

Data Structures

- maybe
- result
- validation

Module folktale/validation

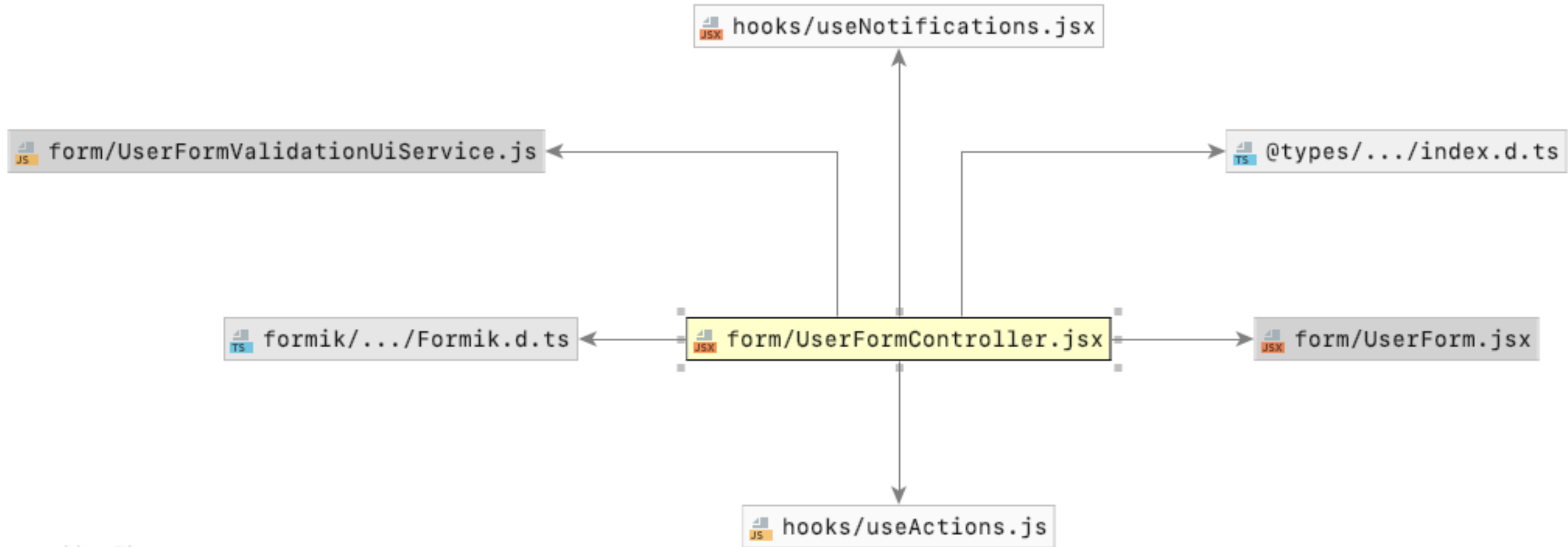
- `Success(value)`
- `Failure(error)`
- Combining validations with `collect()`
- Transforming values with `map()`, `mapFailure()`
- Reacting to results with `matchWith()`

Example

Libraries and Tools

- ramda
- material-ui
- formik-material-ui
- notistack
- axios
- Mockoon

Structure



Powered by yFiles

Thank you!