

ToiletPaper #141

Interaktive Formulare mit Formik

Autor: Kemal Kaya / Software Engineer / Business Division New Business

✗ Problem

Interaktive Formulare sollten folgende Anforderungen erfüllen:

- Input-Felder sollen validiert werden.
- Der Benutzer soll Feedback bekommen.
- Während das Formular abgeschickt wird, könnte der Submit-Button ebenfalls deaktiviert sein.

Natürlich können wir diese Anforderungen mit React-Bordmitteln realisieren. Jedoch müssen wir dafür in React viel Code schreiben. Wir könnten zwar vieles abstrahieren, aber warum das Rad neu erfinden?

✓ Lösung

Formik kann uns beim Entwickeln unserer Formulare unterstützen, indem es uns die wiederkehrenden Aufgaben abnimmt (z. B. Formularzustand, Validierung, Fehlerbehandlung).

Die Bibliothek ist eine populäre Lösung und wird auch in der Dokumentation von React als Fully-Fledged-Solution erwähnt – siehe unten. Es hat eine überschaubare API und lässt sich sehr einfach in Projekte einbinden.

➔ Beispiel

Beispielformular mit Formik und Yup als optionale Hilfsbibliothek (import-Anweisungen sind aus Platzgründen ausgelassen)

```
const formSchema = Yup.object({
  username: Yup.string().min(2, "At least 2 characters").required("Required"),
});

type FormValues = Yup.InferType<typeof formSchema>;
type Props = { initialValues: FormValues };

export const FormikForm = ({ initialValues }: Props) => {
  const formik = useFormik<FormValues>({
    initialValues,
    validationSchema: formSchema,
    onSubmit: (v: FormValues) => { setTimeout(() => setSubmitting(false), 900); }
  });
  const { isSubmitting, errors, touched, values, setSubmitting,
    handleBlur, handleChange, handleSubmit } = formik;

  return (
    <form onSubmit={handleSubmit}>
      <label>
        Username
        <input name="username"
          value={values.username}
          onBlur={handleBlur}
          onChange={handleChange} />
      </label>
      {touched.username && errors.username && (
        <span aria-live="polite">{errors.username}</span>)}
      <button type="submit" disabled={isSubmitting}>Send</button>
    </form>
  );
};
```

+ Weiterführende Aspekte

- <https://formik.org/docs/overview> – Dokumentation von Formik
- <https://reactjs.org/docs/forms.html> – Formik als Fully-Fledged-Solution