

Formik

React 表单增强

1. Fomik 简介

1.1 Formik 介绍

增强表单处理能力. 简化表单处理流程.

[官网](<https://jaredpalmer.com/formik/>)

1. Fomik 简介

1.2 Formik 下载

```
npm install formik
```

2. Formik 增强表单

2.1 Formik 基本使用

使用 formik 进行表单数据绑定以及表单提交处理.

```
import { useFormik } from 'formik';

function App () {
  const formik = useFormik({initialValues: { username: '张三' }, onSubmit: values => {} });
  return <form onSubmit={formik.handleSubmit}>
    <input type="text" name="username"
      value={formik.values.username}
      onChange={formik.handleChange}/>
    <input type="submit"/>
  </form>
}
```

Formik 表单验证

2. Formik 增强表单

2.2 表单验证

2.2.1 初始验证方式

```
const formik = useFormik({  
  validate: values => {  
    const errors = {};  
    if (!values.username) errors.username = '请输入用户名';  
    return errors;  
  }  
});  
return <form>{formik.errors.username ? <div>{formik.errors.username}</div> : null}</form>
```

Formik 表单验证体验优化

2. Formik 增强表单

2.2 表单验证

2.2.2 完善错误信息提示时的用户体验

1. 开启离开焦点时触发验证

```
onBlur={formik.handleBlur}
```


2. Formik 增强表单

2.2 表单验证

2.2.2 完善错误信息提示时的用户体验

2. 提示信息时检查表单元素的值是否被改动过

```
{formik.touched.username && formik.errors.username ? <div>{formik.errors.username}</div> : null}
```

表单验证 Yup

2. Formik 增强表单

2.2 表单验证

2.2.3 使用 yup 验证

1. 下载 yup

```
npm install yup
```

2. Formik 增强表单

2.2 表单验证

2.2.3 使用 yup 验证

2. 引入 yup

```
import * as Yup from 'yup';
```

2. Formik 增强表单

2.2 表单验证

2.2.3 使用 yup 验证

3. 定义验证规则

```
validationSchema: Yup.object({  
  username: Yup.string()  
    .max(15, '用户名的长度不能大于15')  
    .required('请输入用户名'),  
  password: Yup.string()  
    .max(20, '密码的长度不能大于20')  
    .required('请输入密码')  
})
```

getFieldProps 方法

2. Formik 增强表单

2.3 减少样板代码

```
{...formik.getFieldProps('username')}
```

使用组件构建表单

2. Formik 增强表单

2.4 使用组件的方式构建表单

```
import { Formik, Form, Field, ErrorMessage } from "formik";

function App() {
  return (
    <Formik
      initialValues={initialValues}
      onSubmit={handleSubmit}
      validationSchema={schema}
    >
      <Form>
        <Field name="username" />
        <ErrorMessage name="username" />
        <button type="submit">提交</button>
      </Form>
    </Formik>
  );
}
```

Field 组件的 as 属性

2. Formik 增强表单

2.5 构建其他表单项

默认情况下, Field组件渲染的是文本框. 如要生成其他表单元素可以使用以下语法.

```
<Field name="content" as="textarea"/>
<Field name="subject" as="select">
  <option value="前端">前端</option>
  <option value="Java">Java</option>
  <option value="python">python</option>
</Field>
```

构建自定义表单控件

2. Formik 增强表单

2.6 使用 useField 构建 自定义表单控件

```
import { useField } from "formik";

function MyInputField ({label, ...props}) {
  const [field, meta] = useField(props)
  return <div>
    <label htmlFor={props.id}>{label}</label>
    <input {...field} {...props}/>
    {meta.touched && meta.error ? <div>meta.error</div> : null}
  </div>
}

<MyInputField label="密码" type="password" name="password" placeholder="请输入密码" />
```

自定义表单控件 Checkbox