import React, { Component } from "react";

import { Input, Button } from "antd";

// 扩展表单的高阶组件，提供输入控件包装、事件处理、表单校验等

function kFormCreate(Comp) {

return class extends React.Component {

constructor(props) {

super(props);

this.options = {}; // 各字段选项

this.state = {}; // 各字段值

}

handleChange = e => {

let { name, value } = e.target;

this.setState(

{

// 通过计算属性动态指定要修改字段

[name]: value

},

() => {

// 校验

this.validateField(name);

}

);

};

// 包装输入控件、事件处理，field字段名，options选项，InputComp输入控件

getFieldDec = (field, option, InputComp) => {

this.options[field] = option;

return (

<div>

{/\* 由React.createElement生成的元素不能修改，需要克隆一份再扩展 \*/}

{React.cloneElement(InputComp, {

name: field, // 控件name

value: this.state[field] || '', // 控件值

onChange: this.handleChange, // 控件change事件处理

})}

{this.state[field + "Message"] && (

<p style={{ color: "red" }}>{this.state[field + "Message"]}</p>

)}

</div>

);

};

// 校验所有字段

validate = cb => {

// 将选项中所有field组成的数组转换为它们校验结果数组

const rets = Object.keys(this.options).map(field => {

return this.validateField(field);

});

// 校验结果中每一项都要求true

const ret = rets.every(v => v==true);

cb(ret, this.state);

};

// 校验指定字段

validateField = field => {

let rules = this.options[field].rules;

// 只要有任何一项校验失败就返回true并跳出

let ret = rules.some(rule => {

if (rule.required) {

// 仅验证必填项

if (!this.state[field]) {

// 校验失败

this.setState({

[field + "Message"]: rule.message

});

return true;

}

}

});

// 没有失败，清除错误信息

if (!ret) {

this.setState({

[field + "Message"]: ""

});

}

return !ret; // 对返回值取反，表示校验结果

};

render() {

return <Comp {...this.props} getFieldDec={this.getFieldDec} validate={this.validate}/>;

}

};

}

@kFormCreate

class KFormTest extends Component {

onSubmit = e => {

e.preventDefault();

// 校验

this.props.validate((isValid, data) => {

if (isValid) {

console.log('提交登录', data);

} else {

alert('校验失败')

}

})

};

render() {

// 结构出扩展的方法

const { getFieldDec } = this.props;

return (

<div>

{getFieldDec(

"uname",

{ rules: [{ required: true, message: "请输入用户名" }], value: 'jerry' },

<input type="text" />

)}

{getFieldDec(

"pwd",

{ rules: [{ required: true, message: "请输入密码" }] },

<input type="password" />

)}

<button onClick={this.onSubmit}>登录</button>

</div>

);

}

}

export default KFormTest;