

zh-CN

通过 `onFieldsChange` 和 `fields` , 可以把表单的数据存储到上层组件或者 [Redux](#)、[dva](#) 中, 更多可参考 [rc-field-form 示例](#)。

注意: 将表单数据存储于外部容器 [并非好的实践](#), 如无必要请避免使用。

en-US

We can store form data into upper component or [Redux](#) or [dva](#) by using `onFieldsChange` and `fields` , see more at this [rc-field-form demo](#).

Note: Save Form data globally [is not a good practice](#). You should avoid this if not necessary.

```
import React, { useState } from 'react';
import { Form, Input } from 'antd';

interface FieldData {
  name: string | number | (string | number)[];
  value?: any;
  touched?: boolean;
  validating?: boolean;
  errors?: string[];
}

interface CustomizedFormProps {
  onChange: (fields: FieldData[]) => void;
  fields: FieldData[];
}

const CustomizedForm: React.FC<CustomizedFormProps> = ({ onChange, fields }) => (
  <Form
    name="global_state"
    layout="inline"
    fields={fields}
    onFieldsChange={(_, allFields) => {
      onChange(allFields);
    }}
  >
    <Form.Item
      name="username"
      label="Username"
      rules={[{ required: true, message: 'Username is required!' }]}
    >
      <Input />
    </Form.Item>
  </Form>
);

const Demo = () => {
  const [fields, setFields] = useState<FieldData[]>([
    { name: ['username'], value: 'Ant Design' }
  ]);
```

```
return (  
  <>  
    <CustomizedForm  
      fields={fields}  
      onChange={newFields => {  
        setFields(newFields);  
      }}  
    />  
    <pre className="language-bash">{JSON.stringify(fields, null, 2)}</pre>  
  </>  
);  
};  
  
export default () => <Demo />;
```