zh-CN

通过 `Form.Provider` 在表单间处理数据。本例子中，Modal 的确认按钮在 Form 之外，通过 `form.submit` 方法调用表单提交功能。反之，则推荐使用 `<Button htmlType="submit" />` 调用 web 原生提交逻辑。

en-US

Use `Form.Provider` to process data between forms. In this case, submit button is in the Modal which is out of Form. You can use `form.submit` to submit form. Besides, we recommend native `<Button htmlType="submit" />` to submit a form.

```
import React, { useState, useEffect, useRef } from 'react';
import { Form, Input, InputNumber, Modal, Button, Avatar, Typography } from 'antd';
import { SmileOutlined, UserOutlined } from '@ant-design/icons';
import { FormInstance } from 'antd/lib/form';

const layout = {
  labelCol: { span: 8 },
  wrapperCol: { span: 16 },
};
const tailLayout = {
  wrapperCol: { offset: 8, span: 16 },
};

interface UserType {
  name: string;
  age: string;
}

interface ModalFormProps {
  visible: boolean;
  onCancel: () => void;
}

// reset form fields when modal is form, closed
const useResetFormOnCloseModal = ({ form, visible }: { form: FormInstance; visible:
boolean }) => {
  const prevVisibleRef = useRef<boolean>();
  useEffect(() => {
    prevVisibleRef.current = visible;
  }, [visible]);
  const prevVisible = prevVisibleRef.current;

  useEffect(() => {
    if (!visible && prevVisible) {
      form.resetFields();
    }
  }, [visible]);
};

const ModalForm: React.FC<ModalFormProps> = ({ visible, onCancel }) => {
  const [form] = Form.useForm();

  useResetFormOnCloseModal({
    form,
    visible,
  });

  const onOk = () => {
    form.submit();
  };

  return (
    <Modal title="Basic Drawer" visible={visible} onOk={onOk} onCancel={onCancel}>
      <Form form={form} layout="vertical" name="userForm">
        <Form.Item name="name" label="User Name" rules={[{ required: true }]}>
          <Input />
        </Form.Item>
        <Form.Item name="age" label="User Age" rules={[{ required: true }]}>
          <InputNumber />
        </Form.Item>
      </Form>
    </Modal>
  );
};

const Demo = () => {
  const [visible, setVisible] = useState(false);

  const showUserModal = () => {
    setVisible(true);
  };

  const hideUserModal = () => {
    setVisible(false);
  };

  const onFinish = (values: any) => {
    console.log('Finish:', values);
  };

  return (
    <Form.Provider
      onFormFinish={(name, { values, forms }) => {
        if (name === 'userForm') {
          const { basicForm } = forms;
          const users = basicForm.getFieldValue('users') || [];
          basicForm.setFieldsValue({ users: [...users, values] });
          setVisible(false);
        }
      }}
    >
      <Form {...layout} name="basicForm" onFinish={onFinish}>
```

```jsx
        <Form.Item name="group" label="Group Name" rules={[{ required: true }]}>
          <Input />
        </Form.Item>
        <Form.Item
          label="User List"
          shouldUpdate={(prevValues, curValues) => prevValues.users !==
curValues.users}
        >
          {({ getFieldValue }) => {
            const users: UserType[] = getFieldValue('users') || [];
            return users.length ? (
              <ul>
                {users.map((user, index) => (
                  <li key={index} className="user">
                    <Avatar icon={<UserOutlined />} />
                    {user.name} - {user.age}
                  </li>
                ))}
              </ul>
            ) : (
              <Typography.Text className="ant-form-text" type="secondary">
                ( <SmileOutlined /> No user yet. )
              </Typography.Text>
            );
          }}
        </Form.Item>
        <Form.Item {...tailLayout}>
          <Button htmlType="submit" type="primary">
            Submit
          </Button>
          <Button htmlType="button" style={{ margin: '0 8px' }} onClick=
{showUserModal}>
            Add User
          </Button>
        </Form.Item>
      </Form>

      <ModalForm visible={visible} onCancel={hideUserModal} />
    </Form.Provider>
  );
};

export default () => <Demo />;
```

```css
#components-form-demo-form-context .user {
  margin-bottom: 8px;
}

#components-form-demo-form-context .user .ant-avatar {
  margin-right: 8px;
}
```

```
.ant-row-rtl #components-form-demo-form-context .user .ant-avatar {
  margin-right: 0;
  margin-left: 8px;
}
```