


zh-CN

带行编辑功能的表格。

 想要 3 分钟实现? 试试 [ProTable 的可编辑表格!](#)

en-US

Table with editable rows.

```
import React, { useState } from 'react';
import { Table, Input, InputNumber, Popconfirm, Form, Typography } from 'antd';

interface Item {
  key: string;
  name: string;
  age: number;
  address: string;
}

const originData: Item[] = [];
for (let i = 0; i < 100; i++) {
  originData.push({
    key: i.toString(),
    name: `Edrward ${i}`,
    age: 32,
    address: `London Park no. ${i}`,
  });
}

interface EditableCellProps extends React.HTMLAttributes<HTMLElement> {
  editing: boolean;
  dataIndex: string;
  title: any;
  inputType: 'number' | 'text';
  record: Item;
  index: number;
  children: React.ReactNode;
}

const EditableCell: React.FC<EditableCellProps> = ({
  editing,
  dataIndex,
  title,
  inputType,
  record,
  index,
  children,
  ...restProps
}) => {
  const inputNode = inputType === 'number' ? <InputNumber /> : <Input />;

  return (
```

```

<td {...restProps}>
  {editing ? (
    <Form.Item
      name={dataIndex}
      style={{ margin: 0 }}
      rules={[
        {
          required: true,
          message: `Please Input ${title}!`,
        },
      ]}
    >
      {inputNode}
    </Form.Item>
  ) : (
    children
  )}
</td>
);
};

const EditableTable = () => {
  const [form] = Form.useForm();
  const [data, setData] = useState(originData);
  const [editingKey, setEditingKey] = useState('');

  const isEditing = (record: Item) => record.key === editingKey;

  const edit = (record: Partial<Item> & { key: React.Key }) => {
    form.setFieldsValue({ name: '', age: '', address: '', ...record });
    setEditingKey(record.key);
  };

  const cancel = () => {
    setEditingKey('');
  };

  const save = async (key: React.Key) => {
    try {
      const row = (await form.validateFields()) as Item;

      const newData = [...data];
      const index = newData.findIndex(item => key === item.key);
      if (index > -1) {
        const item = newData[index];
        newData.splice(index, 1, {
          ...item,
          ...row,
        });
        setData(newData);
        setEditingKey('');
      } else {

```

```

        newData.push(row);
        setData(newData);
        setEditingKey('');
    }
} catch (errInfo) {
    console.log('Validate Failed:', errInfo);
}
};

const columns = [
    {
        title: 'name',
        dataIndex: 'name',
        width: '25%',
        editable: true,
    },
    {
        title: 'age',
        dataIndex: 'age',
        width: '15%',
        editable: true,
    },
    {
        title: 'address',
        dataIndex: 'address',
        width: '40%',
        editable: true,
    },
    {
        title: 'operation',
        dataIndex: 'operation',
        render: (_, any, record: Item) => {
            const editable = isEditing(record);
            return editable ? (
                <span>
                    <Typography.Link onClick={() => save(record.key)} style={{ marginRight:
8 }}>
                        Save
                    </Typography.Link>
                    <Popconfirm title="Sure to cancel?" onConfirm={cancel}>
                        <a>Cancel</a>
                    </Popconfirm>
                </span>
            ) : (
                <Typography.Link disabled={editingKey !== ''} onClick={() =>
edit(record)}>
                    Edit
                </Typography.Link>
            );
        },
    },
];

```

```

const mergedColumns = columns.map(col => {
  if (!col.editable) {
    return col;
  }
  return {
    ...col,
    onCell: (record: Item) => ({
      record,
      inputType: col.dataIndex === 'age' ? 'number' : 'text',
      dataIndex: col.dataIndex,
      title: col.title,
      editing: isEditing(record),
    }),
  };
});

return (
  <Form form={form} component={false}>
    <Table
      components={{
        body: {
          cell: EditableCell,
        },
      }}
      bordered
      dataSource={data}
      columns={mergedColumns}
      rowClassName="editable-row"
      pagination={{
        onChange: cancel,
      }}
    />
  </Form>
);
};

export default () => <EditableTable />;

```

```

.editable-row .ant-form-item-explain {
  position: absolute;
  top: 100%;
  font-size: 12px;
}

```