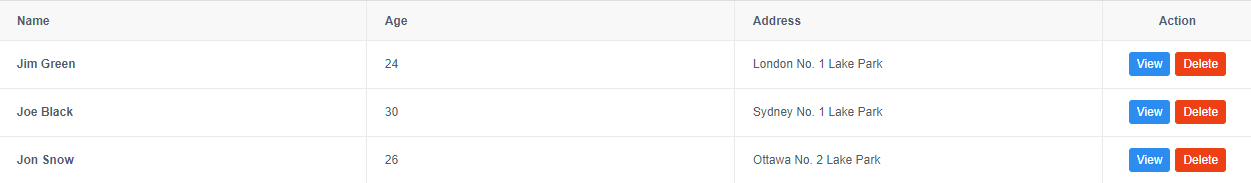
# 使用render函数完成渲染的table效果

Table表格

关于生成自定义列的表格，效果如下：

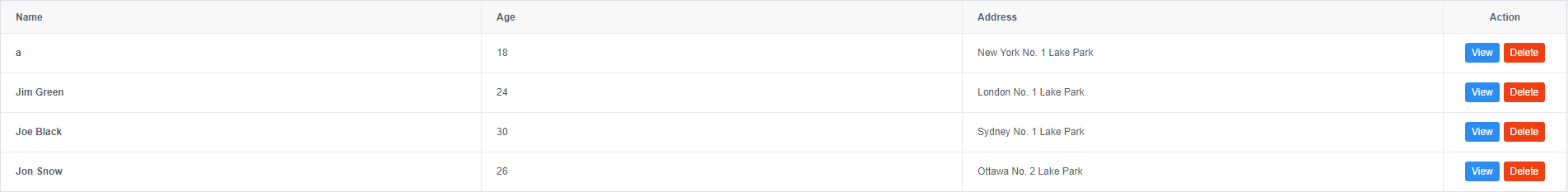


代码如下：

<**template**>  
 <**Table** border :columns="columns7" :data="data6"></**Table**>  
</**template**>  
<**script**>  
 **export default** {  
 data () {  
 **return** {  
 columns7: [  
 {  
 title: 'Name',  
 key: 'name',  
 render: (h, params) => {  
 **return** h('div', [  
 h('Icon', {  
 props: {  
 type: 'person'//此处person就是决定在下面定义渲染的<strong>标签中的<i>标签里的class属性的值，  
 // 如这里是ivu-icon ivu-icon-person。如过type定义成p那么class属性的值就是ivu-icon ivu-icon-p  
 }  
 }),  
 h('strong', params.row.name)  
 ]);  
 }  
 },  
 {  
 title: 'Age',  
 key: 'age'  
 },  
 {  
 title: 'Address',  
 key: 'address'  
 },  
 {  
 title: 'Action',  
 key: 'action',  
 width: 150,  
 align: 'center',  
 render: (h, params) => {  
 **return** h('div', [  
 h('Button', {  
 props: {  
 //type: 'primary',//这里type是决定渲染生成的button标签的样式采用iview中的样式（相当于就是生成哪种iview中的button）,在这里我设置成了自定义的，也就是下面Style中的  
 size: 'small'  
 },  
 style: {  
 marginRight: '5px',  
 color:'#fff', //设置字体的颜色为白色  
 backgroundColor:'#67ed14',//设置背景颜色为绿色  
 borderColor:'#440101'//设置边框的颜色  
 },  
 on: {  
 click: () => {  
 **this**.show(params.index)  
 }  
 }  
 }, 'View'),  
 h('Button', {  
 props: {  
 type: 'error',//此处也是采用了iview中的样式  
 size: 'small'  
 },  
 on: {  
 click: () => {  
 **this**.remove(params.index)  
 }  
 }  
 }, 'Delete')  
 ]);  
 }  
 }  
 ],  
 data6: [  
 {  
 name: 'John Brown',  
 age: 18,  
 address: 'New York No. 1 Lake Park'  
 },  
 {  
 name: 'Jim Green',  
 age: 24,  
 address: 'London No. 1 Lake Park'  
 },  
 {  
 name: 'Joe Black',  
 age: 30,  
 address: 'Sydney No. 1 Lake Park'  
 },  
 {  
 name: 'Jon Snow',  
 age: 26,  
 address: 'Ottawa No. 2 Lake Park'  
 }  
 ]  
 }  
 },  
 methods: {  
 show (index) {  
 **this**.$Modal.info({  
 title: 'User Info',  
 content: `Name：${**this**.data6[index].name}<br>Age：${**this**.data6[index].age}<br>Address：${**this**.data6[index].address}`  
 })  
 },  
 remove (index) {  
 **this**.data6.splice(index, 1);  
 }  
 }  
 }  
</**script**>

**注意：上面h(tagName,())或者h(tagName,[])的写法就等同于render函数的createElenent(tagName,())或createElement(tagName,[])的写法**

# 关于使用render函数完成类似1代表男，0代表女的过滤效果的table

主要代码：

