

zh-CN

通过 `filterDropdown` 自定义的列筛选功能，并实现一个搜索列的示例。

给函数 `clearFilters` 添加 `boolean` 类型参数 `closeDropdown`，是否关闭筛选菜单，默认为 `true`。添加 `boolean` 类型参数 `confirm`，清除筛选时是否提交已选项，默认 `true`。

en-US

Implement a customized column search example via `filterDropdown`.

Add the `boolean` type parameter `closeDropdown` to the function `clearFilters`. Whether to close the filter menu is `true` by default. Add the `boolean` type parameter `confirm` to clear whether to submit the option during filtering. The default is `true`.

```
import { Table, Input, Button, Space } from 'antd';
import Highlighter from 'react-highlight-words';
import { SearchOutlined } from '@ant-design/icons';

const data = [
  {
    key: '1',
    name: 'John Brown',
    age: 32,
    address: 'New York No. 1 Lake Park',
  },
  {
    key: '2',
    name: 'Joe Black',
    age: 42,
    address: 'London No. 1 Lake Park',
  },
  {
    key: '3',
    name: 'Jim Green',
    age: 32,
    address: 'Sidney No. 1 Lake Park',
  },
  {
    key: '4',
    name: 'Jim Red',
    age: 32,
    address: 'London No. 2 Lake Park',
  },
];

class App extends React.Component {
  state = {
    searchText: '',
    searchedColumn: '',
  };
}
```

```

getColumnSearchProps = dataIndex => ({
  filterDropdown: ({ setSelectedKeys, selectedKeys, confirm, clearFilters }) => (
    <div style={{ padding: 8 }}>
      <Input
        ref={node => {
          this.searchInput = node;
        }}
        placeholder={`Search ${dataIndex}`}
        value={selectedKeys[0]}
        onChange={e => setSelectedKeys(e.target.value ? [e.target.value] : [])}
        onPressEnter={() => this.handleSearch(selectedKeys, confirm, dataIndex)}
        style={{ marginBottom: 8, display: 'block' }}
      />
      <Space>
        <Button
          type="primary"
          onClick={() => this.handleSearch(selectedKeys, confirm, dataIndex)}
          icon={<SearchOutlined />}
          size="small"
          style={{ width: 90 }}
        >
          Search
        </Button>
        <Button onClick={() => this.handleReset(clearFilters)} size="small" style=
{{ width: 90 }}>
          Reset
        </Button>
        <Button
          type="link"
          size="small"
          onClick={() => {
            confirm({ closeDropdown: false });
            this.setState({
              searchText: selectedKeys[0],
              searchedColumn: dataIndex,
            });
          }}
        >
          Filter
        </Button>
      </Space>
    </div>
  ),
  filterIcon: filtered => <SearchOutlined style={{ color: filtered ? '#1890ff' :
undefined }} />,
  onFilter: (value, record) =>
    record[dataIndex]
      ? record[dataIndex].toString().toLowerCase().includes(value.toLowerCase())
      : '',
  onFilterDropdownVisibleChange: visible => {
    if (visible) {

```

```

        setTimeout(() => this.searchInput.select(), 100);
    }
},
render: text =>
    this.state.searchedColumn === dataIndex ? (
        <Highlighter
            highlightStyle={{ backgroundColor: '#ffc069', padding: 0 }}
            searchWords={[this.state.searchText]}
            autoEscape
            textToHighlight={text ? text.toString() : ''}
        />
    ) : (
        text
    ),
});

handleSearch = (selectedKeys, confirm, dataIndex) => {
    confirm();
    this.setState({
        searchText: selectedKeys[0],
        searchedColumn: dataIndex,
    });
};

handleReset = clearFilters => {
    clearFilters();
    this.setState({ searchText: '' });
};

render() {
    const columns = [
        {
            title: 'Name',
            dataIndex: 'name',
            key: 'name',
            width: '30%',
            ...this.getColumnSearchProps('name'),
        },
        {
            title: 'Age',
            dataIndex: 'age',
            key: 'age',
            width: '20%',
            ...this.getColumnSearchProps('age'),
        },
        {
            title: 'Address',
            dataIndex: 'address',
            key: 'address',
            ...this.getColumnSearchProps('address'),
            sorter: (a, b) => a.address.length - b.address.length,
            sortDirections: ['descend', 'ascend'],
        },
    ];

```

```
    },  
  ],  
  return <Table columns={columns} dataSource={data} />;  
}  
}  
  
export default () => <App />;
```