## zh-CN

可搜索的树。

## en-US

Searchable Tree.

```
import { Tree, Input } from 'antd';

const { Search } = Input;

const x = 3;
const y = 2;
const z = 1;
const gData = [];

const generateData = (_level, _preKey, _tns) => {
  const preKey = _preKey || '0';
  const tns = _tns || gData;

  const children = [];
  for (let i = 0; i < x; i++) {
    const key = `${preKey}-${i}`;
    tns.push({ title: key, key });
    if (i < y) {
      children.push(key);
    }
  }
  if (_level < 0) {
    return tns;
  }
  const level = _level - 1;
  children.forEach((key, index) => {
    tns[index].children = [];
    return generateData(level, key, tns[index].children);
  });
};
generateData(z);

const dataList = [];
const generateList = data => {
  for (let i = 0; i < data.length; i++) {
    const node = data[i];
    const { key } = node;
    dataList.push({ key, title: key });
    if (node.children) {
      generateList(node.children);
    }
  }
};
```

```
generateList(gData);

const getParentKey = (key, tree) => {
  let parentKey;
  for (let i = 0; i < tree.length; i++) {
    const node = tree[i];
    if (node.children) {
      if (node.children.some(item => item.key === key)) {
        parentKey = node.key;
      } else if (getParentKey(key, node.children)) {
        parentKey = getParentKey(key, node.children);
      }
    }
  }
  return parentKey;
};

class SearchTree extends React.Component {
  state = {
    expandedKeys: [],
    searchValue: '',
    autoExpandParent: true,
  };

  onExpand = expandedKeys => {
    this.setState({
      expandedKeys,
      autoExpandParent: false,
    });
  };

  onChange = e => {
    const { value } = e.target;
    const expandedKeys = dataList
      .map(item => {
        if (item.title.indexOf(value) > -1) {
          return getParentKey(item.key, gData);
        }
        return null;
      })
      .filter((item, i, self) => item && self.indexOf(item) === i);
    this.setState({
      expandedKeys,
      searchValue: value,
      autoExpandParent: true,
    });
  };

  render() {
    const { searchValue, expandedKeys, autoExpandParent } = this.state;
    const loop = data =>
      data.map(item => {
```

```
        const index = item.title.indexOf(searchValue);
        const beforeStr = item.title.substring(0, index);
        const afterStr = item.title.slice(index + searchValue.length);
        const title =
          index > -1 ? (
            <span>
              {beforeStr}
              <span className="site-tree-search-value">{searchValue}</span>
              {afterStr}
            </span>
          ) : (
            <span>{item.title}</span>
          );
        if (item.children) {
          return { title, key: item.key, children: loop(item.children) };
        }

        return {
          title,
          key: item.key,
        };
      });
    return (
      <div>
        <Search style={{ marginBottom: 8 }} placeholder="Search" onChange=
{this.onChange} />
        <Tree
          onExpand={this.onExpand}
          expandedKeys={expandedKeys}
          autoExpandParent={autoExpandParent}
          treeData={loop(gData)}
        />
      </div>
    );
  }
}

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

```
.site-tree-search-value {
  color: #f50;
}
```