



链滴

React Visual - 树视图组件 (advanced)

作者: [Vanessa](#)

原文链接: <https://ld246.com/article/1571625428308>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

描述

使用 JSON 或数组渲染一个内容可折叠的树视图。

- 使用对象解构为特定的属性设置默认值
- 使用 `toggle` 值来确定内容的初始化状态（展开/收拢）
- 使用 `React.useState()` hook 创建 `isToggled` 状态变量，设置初始值为 `toggle` 属性
- 返回一个 `<div>` 元素，他包裹着内容组件和用于切换组件 `isToggled` 状态的 `` 元素
- 基于 `isParentToggled`, `isToggled`, `name` 及 `Array.isArray()` 对 `data` 的判断结果来设定组件展现
- 循环 `data` 中的每一个节点，如果为对象或数组就使用递归将其渲染为一个子树
- 否则的话，使用适当的样式将其渲染为一个 `<p>` 元素

实现

```
.tree-element {  
  margin: 0;  
  position: relative;  
}
```

```
div.tree-element:before {  
  content: " ";  
  position: absolute;  
  top: 24px;  
  left: 1px;  
  height: calc(100% - 48px);  
  border-left: 1px solid gray;  
}
```

```
.toggler {  
  position: absolute;  
  top: 10px;  
  left: 0px;  
  width: 0;  
  height: 0;  
  border-top: 4px solid transparent;  
  border-bottom: 4px solid transparent;  
  border-left: 5px solid gray;  
  cursor: pointer;  
}
```

```
.toggler.closed {  
  transform: rotate(90deg);  
}
```

```
.collapsed {  
  display: none;
```

```

}

function TreeView({
  data,
  toggled = true,
  name = null,
  isLast = true,
  isChildElement = false,
  isParentToggled = true
}) {
  const [isToggled, setIsToggled] = React.useState(toggled);

  return (
    <div
      style={{ marginLeft: isChildElement ? 16 : 4 + 'px' }}
      className={isParentToggled ? 'tree-element' : 'tree-element collapsed'}
    >
      <span
        className={isToggled ? 'toggler' : 'toggler closed'}
        onClick={() => setIsToggled(!isToggled)}
      />
      {name ? <strong>&nbsp;&nbsp;&nbsp;</strong> : <span>&nbsp;&nbsp;&</span>}
      {Array.isArray(data) ? '[' : '{'}
      {!isToggled && '...'}
      {Object.keys(data).map((v, i, a) =>
        typeof data[v] === 'object' ? (
          <TreeView
            data={data[v]}
            isLast={i === a.length - 1}
            name={Array.isArray(data) ? null : v}
            isChildElement
            isParentToggled={isParentToggled && isToggled}
          />
        ) : (
          <p
            style={{ marginLeft: 16 + 'px' }}
            className={isToggled ? 'tree-element' : 'tree-element collapsed'}
          >
            {Array.isArray(data) ? " : <strong>{v}: </strong>}
            {data[v]}
            {i === a.length - 1 ? " : ','}
          </p>
        )
      )}
      {Array.isArray(data) ? ']' : '}' }
      {!isLast ? ',' : ''}
    </div>
  );
}

```

使用

```

let data = {
  lorem: {

```

```
    ipsum: 'dolor sit',
    amet: {
      consectetur: 'adipiscing',
      elit: [
        'duis',
        'vitae',
        {
          semper: 'orci'
        },
        {
          est: 'sed ornare'
        },
        'etiam',
        ['laoreet', 'tincidunt'],
        ['vestibulum', 'ante']
      ]
    },
    ipsum: 'primis'
  }
};
ReactDOM.render(<TreeView data={data} name="data" />, document.getElementById('root'))
```

返回总目录

[每天 30 秒系列之 React](#)