

多层次的结构列表。

何时使用

文件夹、组织架构、生物分类、国家地区等等，世间万物的大多数结构都是树形结构。使用 `树控件` 可以完整展现其中的层级关系，并具有展开收起选择等交互功能。

API

Tree props

参数	说明	类型	默认值	版本
<code>allowDrop</code>	是否允许拖拽时放置在该节点	<code>(({ dropNode, dropPosition }) => boolean</code>	-	
<code>autoExpandParent</code>	是否自动展开父节点	<code>boolean</code>	<code>false</code>	
<code>blockNode</code>	是否节点占据一行	<code>boolean</code>	<code>false</code>	
<code>checkable</code>	节点前添加 <code>Checkbox</code> 复选框	<code>boolean</code>	<code>false</code>	
<code>checkedKeys</code>	（受控）选中复选框的树节点 （注意：父子节点有关联，如果传入父节点 <code>key</code> ，则子节点自动选中；相应当子节点 <code>key</code> 都传入，父节点也自动选中。当设置 <code>checkable</code> 和 <code>checkStrictly</code> ，它是一个有 <code>checked</code> 和 <code>halfChecked</code> 属性的对象，并且父子节点的选中与否不再关联	<code>string[] {checked: string[], halfChecked: string[]}</code>	<code>[]</code>	
<code>checkStrictly</code>	<code>checkable</code> 状态下节点选择完全受控（父子节点选中状态不再关联）	<code>boolean</code>	<code>false</code>	
<code>defaultCheckedKeys</code>	默认选中复选框的树节点	<code>string[]</code>	<code>[]</code>	
<code>defaultExpandAll</code>	默认展开所有树节点	<code>boolean</code>	<code>false</code>	
<code>defaultExpandedKeys</code>	默认展开指定的树节点	<code>string[]</code>	<code>[]</code>	
<code>defaultExpandParent</code>	默认展开父节点	<code>boolean</code>	<code>true</code>	
<code>defaultSelectedKeys</code>	默认选中的树节点	<code>string[]</code>	<code>[]</code>	
<code>disabled</code>	将树禁用	<code>boolean</code>	<code>false</code>	
<code>draggable</code>	设置节点可拖拽，可以通过 <code>icon: false</code> 关闭拖拽提示图标	<code>boolean ((node: DataNode) => boolean) { icon?: React.ReactNode false, nodeDraggable?:</code>	<code>false</code>	<code>config:</code> <code>4.17.0</code>

		(node: DataNode) => boolean }		
expandedKeys	(受控) 展开指定的树节点	string[]	[]	
fieldNames	自定义节点 title、key、children 的字段	object	{ title: title, key: key, children: children }	4.17.0
filterTreeNode	按需筛选树节点（高亮），返回 true	function(node)	-	
height	设置虚拟滚动容器高度，设置后内部节点不再支持横向滚动	number	-	
icon	自定义树节点图标。	ReactNode (props) => ReactNode	-	
loadData	异步加载数据	function(node)	-	
loadedKeys	(受控) 已经加载的节点，需要配合 loadData 使用	string[]	[]	
multiple	支持点选多个节点（节点本身）	boolean	false	
selectable	是否可选中	boolean	true	
selectedKeys	(受控) 设置选中的树节点	string[]	-	
showIcon	是否展示 TreeNode title 前的图标，没有默认样式，如设置为 true，需要自行定义图标相关样式	boolean	false	
showLine	是否展示连接线	boolean {showLeafIcon: boolean}	false	
switcherIcon	自定义树节点的展开/折叠图标	ReactNode	-	
titleRender	自定义渲染节点	(nodeData) => ReactNode	-	4.5.0
treeData	treeNodes 数据，如果设置则不需要手动构造 TreeNode 节点（key 在整个树范围内唯一）	array<{key, title, children, [disabled, selectable]}>	-	
virtual	设置 false 时关闭虚拟滚动	boolean	true	4.1.0
onCheck	点击复选框触发	function(checkedKeys, e:{checked: bool, checkedNodes, node, event, halfCheckedKeys})	-	

onDragEnd	dragend 触发时调用	function({event, node})	-	
onDragEnter	dragenter 触发时调用	function({event, node, expandedKeys})	-	
onDragLeave	dragleave 触发时调用	function({event, node})	-	
onDragOver	dragover 触发时调用	function({event, node})	-	
onDragStart	开始拖拽时调用	function({event, node})	-	
onDrop	drop 触发时调用	function({event, node, dragNode, dragNodesKeys})	-	
onExpand	展开/收起节点时触发	function(expandedKeys, {expanded: bool, node})	-	
onLoad	节点加载完毕时触发	function(loadedKeys, {event, node})	-	
onRightClick	响应右键点击	function({event, node})	-	
onSelect	点击树节点触发	function(selectedKeys, e:{selected: bool, selectedNodes, node, event})	-	

TreeNode props

参数	说明	类型	默认值	
checkable	当树为 checkable 时，设置独立节点是否展示 Checkbox	boolean	-	
disableCheckbox	禁掉 checkbox	boolean	false	
disabled	禁掉响应	boolean	false	
icon	自定义图标。可接收组件，props 为当前节点 props	ReactNode (props) => ReactNode	-	
isLeaf	设置为叶子节点 (设置了 loadData 时有效)。为 false 时会强制将其作为父节点	boolean	-	
key	被树的 (default)ExpandedKeys / (default)CheckedKeys / (default)SelectedKeys 属性所用。注意：整个树范围内的所有节点的 key 值不能重复！	string	(内部计算出的节点位置)	
selectable	设置节点是否可被选中	boolean	true	
title	标题	ReactNode	---	

DirectoryTree props

--	--	--	--

参数	说明	类型	默认值
expandAction	目录展开逻辑，可选：false click doubleClick	string boolean	click

注意

在 3.4.0 之前：树节点可以有很多，但在设置 `checkable` 时，将会花费更多的计算时间，因此我们缓存了一些计算结果（`this.treeNodesStates`）来复用，避免多次重复计算，以此提高性能。但这也带来了一些限制，当你异步加载树节点时，你需要这样渲染树：

```
{
  this.state.treeData.length ? (
    <Tree>
      {this.state.treeData.map(data => (
        <TreeNode />
      ))}
    </Tree>
  ) : (
    'loading tree'
  );
}
```

Tree 方法

名称	说明
<code>scrollTo({ key: string number; align?: 'top' 'bottom' 'auto'; offset?: number })</code>	虚拟滚动下，滚动到指定 key 条目

FAQ

在 showLine 时，如何隐藏子节点图标？

文件图标通过 `switcherIcon` 来实现，如果不需要你可以覆盖对应的样式：<https://codesandbox.io/s/883vo47xp8>

defaultExpandedAll 在异步加载数据时为何不生效？

`default` 前缀属性只有在初始化时生效，因而异步加载数据时 `defaultExpandedAll` 已经执行完成。你可以通过受控 `expandedKeys` 或者在数据加载完成后渲染 `Tree` 来实现全部展开。

虚拟滚动的限制

虚拟滚动通过在仅渲染可视区域的元素来提升渲染性能。但是同时由于不会渲染所有节点，所以无法自动拓转横向宽度（比如超长 `title` 的横向滚动条）。

disabled 节点在树中的关系是什么？

`Tree` 通过传导方式进行数据变更。无论是展开还是勾选，它都会从变更的节点开始向上、向上传导变化，直到遍历的当前节点是 `disabled` 时停止。因而如果控制的节点本身为 `disabled` 时，那么它只会修改本身而不会影响其他节点。举例来说，一个父节点包含 3 个子节点，其中一个为 `disabled` 状态。那么勾选父节点，只会影响其余两个子节点变成勾选状态。勾选两个子节点后，无论 `disabled` 节点什么状态，父节点都会变成勾选状态。

这种传导终止的方式是为了防止通过勾选子节点使得 `disabled` 父节点变成勾选状态，而用户无法直接勾选 `disabled` 父节点更改其状态导致的交互矛盾。如果你有着自己的传导需求，可以通过 `checkStrictly` 自定义勾选逻辑。