Tree

A hierarchical list structure component.

When To Use

Almost anything can be represented in a tree structure. Examples include directories, organization hierarchies, biological classifications, countries, etc. The `Tree` component is a way of representing the hierarchical relationship between these things. You can also expand, collapse, and select a treeNode within a `Tree`.

API

Tree props

Property	Description	Type	Default	Version
allowDrop	Whether to allow dropping on the node	({ dropNode, dropPosition }) => boolean	-	
autoExpandParent	Whether to automatically expand a parent treeNode	boolean	false	
blockNode	Whether treeNode fill remaining horizontal space	boolean	false	
checkable	Add a Checkbox before the treeNodes	boolean	false	

Property	Description	Type	Default	Version
checkedKeys	(Controlled) Specifies the keys of the checked treeNodes (PS: When this specifies the key of a treeNode which is also a parent treeNode, all the children treeNodes of will be checked; and vice versa, when it specifies the key of a treeNode which is a child treeNode, its parent treeNode will also be checked. When `checkable` and `checkStrictly` is true, its object has `checked` and `halfChecked` property. Regardless of whether the child or parent treeNode is checked, they won't impact each other	string[] | {checked: string[], halfChecked: string[]}	[]	

Property	Description	Type	Default	Version
checkStrictly	Check treeNode precisely; parent treeNode and children treeNodes are not associated	boolean	false	
defaultCheckedKeys	Specifies the keys of the default checked treeNodes	string[]	[]	
defaultExpandAll	Whether to expand all treeNodes by default	boolean	false	
defaultExpandedKeys	Specify the keys of the default expanded treeNodes	string[]	[]	
defaultExpandParent	If auto expand parent treeNodes when init	boolean	true	
defaultSelectedKeys	Specifies the keys of the default selected treeNodes	string[]	[]	
disabled	Whether disabled the tree	boolean	false	

Property	Description	Type	Default	Version
draggable	Specifies whether this Tree or the node is draggable. Use `icon: false` to disable drag handler icon	boolean | ((node: DataNode) => boolean) | { icon?: React.ReactNode | false, node-Draggable?: (node: DataNode) => boolean }	false	`config`: 4.17.0
expandedKeys	(Controlled) Specifies the keys of the expanded treeNodes	string[]	[]	
fieldNames	Customize node title, key, children field name	object	{ title: `title`, key: `key`, children: `children` }	4.17.0
filterTreeNode	Defines a function to filter (highlight) treeNodes. When the function returns `true`, the corresponding treeNode will be highlighted	function(node)	-	
height	Config virtual scroll height. Will not support horizontal scroll when enable this	number	-	
icon	Customize treeNode icon	ReactNode | (props) => ReactNode	-	

Property	Description	Type	Default	Version
loadData	Load data asynchronously	function(node)	-	
loadedKeys	(Controlled) Set loaded tree nodes. Need work with `loadData`	string[]	[]	
multiple	Allows selecting multiple treeNodes	boolean	false	
selectable	Whether can be selected	boolean	true	
selectedKeys	(Controlled) Specifies the keys of the selected treeNodes	string[]	-	
showIcon	Shows the icon before a TreeNode's title. There is no default style; you must set a custom style for it if set to true	boolean	false	
showLine	Shows a connecting line	boolean | {showLeafIcon: boolean}	false	
switcherIcon	Customize collapse/expand icon of tree node	ReactNode	-	
titleRender	Customize tree node title render	(nodeData) => ReactNode	-	4.5.0

Property	Description	Type	Default	Version
treeData	The treeNodes data Array, if set it then you need not to construct children TreeNode. (key should be unique across the whole array)	array<{ key, title, children, [disabled, selectable] }>	-	
virtual	Disable virtual scroll when set to false	boolean	true	4.1.0
onCheck	Callback function for when the onCheck event occurs	function(checkedKeys, e:{checked: bool, checkedNodes, node, event, halfChecked-Keys})		
onDragEnd	Callback function for when the onDragEnd event occurs	function({event, node})	-	
onDragEnter	Callback function for when the onDragEnter event occurs	function({event, node, expand-edKeys})	-	
onDragLeave	Callback function for when the onDragLeave event occurs	function({event, node})	-	
onDragOver	Callback function for when the onDragOver event occurs	function({event, node})	-	

Property	Description	Type	Default	Version
onDragStart	Callback function for when the onDragStart event occurs	function({event, node})	-	
onDrop	Callback function for when the onDrop event occurs	function({event, node, dragNode, dragNodesKeys})	-	
onExpand	Callback function for when a treeNode is expanded or collapsed	function(expandedKeys, {expanded: bool, node})	-	
onLoad	Callback function for when a treeNode is loaded	function(loadedKeys, {event, node})	-	
onRightClick	Callback function for when the user right clicks a treeNode	function({event, node})	-	
onSelect	Callback function for when the user clicks a treeNode	function(selectedKeys, e:{selected: bool, selectedNodes, node, event})	-	

TreeNode props

Property	Description	Type	Default
checkable	When Tree is checkable, set TreeNode display Checkbox or not	boolean	-

Property	Description	Type	Default
disableCheckbox	Disables the checkbox of the treeNode	boolean	false
disabled	Disables the treeNode	boolean	false
icon	Customize icon. When you pass component, whose render will receive full TreeNode props as component props	ReactNode | (props) => ReactNode	-
isLeaf	Determines if this is a leaf node(effective when `loadData` is specified). `false` will force trade TreeNode as a parent node	boolean	-
key	Used with (default)ExpandedKeys / (default)CheckedKeys / (default)SelectedKeys. P.S.: It must be unique in all of treeNodes of the tree	string	(internal calculated position of treeNode)
selectable	Set whether the treeNode can be selected	boolean	true
title	Title	ReactNode	---

DirectoryTree props

Property	Description	Type	Default
expandAction	Directory open logic, optional: false | `click` | `doubleClick`	string | boolean	`click`

Note

Before `3.4.0`: The number of treeNodes can be very large, but when `checkable=true`, it will increase the compute time. So, we cache some calculations (e.g. `this.treeNodesStates`) to avoid double computing. But, this brings some restrictions. **When you load treeNodes asynchronously, you should render tree like this**:

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

Tree Methods

Name	Description
scrollTo({ key: string | number; align?: 'top' | 'bottom' | 'auto'; offset?: number })	Scroll to key item in virtual scroll

FAQ

How to hide file icon when use showLine?

File icon realize by using switcherIcon. You can overwrite the style to hide it: https://codesandbox.io/s/883vo47xp8

Why defaultExpandedAll not working on ajax data?

`default` prefix prop only works when inited. So `defaultExpandedAll` has already executed when ajax load data. You can control `expandedKeys` or render Tree when data loaded to realize expanded all.

Virtual scroll limitation

Virtual scroll only render items in visible region. Thus not support auto width (like long `title` with horizontal scroll).

What does `disabled` node work logic in the tree?

Tree change its data by conduction. Includes checked or auto expanded, it will conduction state to parent / children node until current node is `disabled`. So if a controlled node is `disabled`, it will only modify self state and not affect other nodes. For example, a parent node contains 3 child nodes and one of them is `disabled`. When check the parent node, it will only check rest 2 child nodes. As the same, when check these 2 child node, parent will be checked whatever checked state the `disabled` one is.

This conduction logic prevent that modify `disabled` parent checked state by check children node and user can not modify directly with click parent which makes the interactive conflict. If you want to modify this conduction logic, you can customize it with `checkStrictly` prop.