Breadcrumb ✎

A breadcrumb displays the current location within a hierarchy. It allows going back to states higher up in the hierarchy.

When To Use

- When the system has more than two layers in a hierarchy.
- When you need to inform the user of where they are.
- When the user may need to navigate back to a higher level.

```jsx
// works when >=5.3.0, recommended ✅
return <Breadcrumb items={[{ title: 'sample' }]} />;

// works when <5.3.0, deprecated when >=5.3.0 🙅
return (
  <Breadcrumb>
    <Breadcrumb.Item>sample</Breadcrumb.Item>
  </Breadcrumb>
);

// or

return <Breadcrumb routes={[{ breadcrumbName: 'sample' }]} />;
```

Examples

Home / Application Center / Application List /
An Application

Basic Usage ✎

The simplest use.

```tsx
import React from 'react';
import { Breadcrumb } from 'antd';

const App: React.FC = () => (
  <Breadcrumb
    items={[
      {
        title: 'Home',
      },
      {
        title: <a href="">Application Center</a>
      },
      {
        title: <a href="">Application List</a>
      },
      {
        title: 'An Application',
      },
    ]}
  />
);

export default App;
```

⌂ / ⚇ Application List / **Application**

With an Icon ✎

The icon should be placed in front of the text.

```tsx
import { HomeOutlined, UserOutlined } from '@d
import { Breadcrumb } from 'antd';
import React from 'react';

const App: React.FC = () => (
  <Breadcrumb
    items={[
      {
        href: '',
        title: <HomeOutlined />,
      },
      {
        href: '',
        title: (
          <>
            <UserOutlined />
            <span>Application List</span>
          </>
        ),
      },
      {
        title: 'Application',
      },
    ]}
  />
);

export default App;
```

Configuring the Separator ✎

The separator can be customized by setting the separator property: `separator=">"` .

```
react-router V6 ✎

Used together with react-router@6+ .

import React from 'react';
import { Alert, Breadcrumb } from 'antd';
import { HashRouter, Link, Route, Routes, useL

const Apps = () => (
  <ul className="app-list">
    <li>
      <Link to="/apps/1">Application1</Link>:
    </li>
    <li>
      <Link to="/apps/2">Application2</Link>:
    </li>
  </ul>
);

const breadcrumbNameMap: Record<string, string
  '/apps': 'Application List',
  '/apps/1': 'Application1',
  '/apps/2': 'Application2',
  '/apps/1/detail': 'Detail',
  '/apps/2/detail': 'Detail',
};

const Home = () => {
  const location = useLocation();
  const pathSnippets = location.pathname.split

  const extraBreadcrumbItems = pathSnippets.mc
    const url = `/${pathSnippets.slice(0, inde
    return {
      key: url,
      title: <Link to={url}>{breadcrumbNameMar
    };
  });

  const breadcrumbItems = [
    {
      title: <Link to="/">Home</Link>,
      key: 'home',
    },
  ].concat(extraBreadcrumbItems);

  return (
    <div className="demo">
      <div className="demo-nav">
        <Link to="/">Home</Link>
        <Link to="/apps">Application List</Lir
      </div>
      <Routes>
        <Route path="/apps" element={<Apps />]
        <Route path="*" element={<span>Home Pc
      </Routes>
      <Alert style={{ margin: '16px 0' }} mess
      <Breadcrumb items={breadcrumbItems} />
    </div>
  );
};

const App: React.FC = () => (
```

```
import React from 'react';
import { Breadcrumb } from 'antd';

const App: React.FC = () => (
  <Breadcrumb
    separator=">"
    items={[
      {
        title: 'Home',
      },
      {
        title: 'Application Center',
        href: '',
      },
      {
        title: 'Application List',
        href: '',
      },
      {
        title: 'An Application',
      },
    ]}
  />
);

export default App;
```

Bread crumbs with drop down menu ✎

Breadcrumbs support drop down menu.

```
import React from 'react';
import { Breadcrumb } from 'antd';

const menuItems = [
  {
    key: '1',
    label: (
      <a target="_blank" rel="noopener norefer
        General
      </a>
    ),
  },
  {
    key: '2',
    label: (
      <a target="_blank" rel="noopener norefer
        Layout
      </a>
    ),
  },
  {
    key: '3',
    label: (
      <a target="_blank" rel="noopener norefer
        Navigation
      </a>
    ),
  },
];

const App: React.FC = () => (
  <Breadcrumb
    items={[
      {
        title: 'Ant Design',
      },
      {
        title: <a href="">Component</a>,
      },
      {
        title: <a href="">General</a>,
        menu: { items: menuItems },
      },
      {
        title: 'Button',
      },
    ]}
  />
);

export default App;
```

Configuring the Separator Independently ✎

Customize separator for each other.

```
import { Breadcrumb } from 'antd';
import React from 'react';

const App: React.FC = () => (
  <Breadcrumb
    separator=""
    items={[
      {
        title: 'Location',
      },
      {
        type: 'separator',
        separator: ':',
      },
      {
        href: '',
        title: 'Application Center',
      },
      {
        type: 'separator',
      },
      {
        href: '',
        title: 'Application List',
      },
      {
        type: 'separator',
      },
      {
        title: 'An Application',
      },
    ]}
  />
);

export default App;
```

Home / User ⌄

Debug Routes ✎

Origin `routes` debug.

```
import { Breadcrumb } from 'antd';
import React from 'react';

export default () => (
  <Breadcrumb
    routes={[
      {
        path: '/home',
        breadcrumbName: 'Home',
      },
      {
        path: '/user',
        breadcrumbName: 'User',
        children: [
          {
            path: '/user1',
            breadcrumbName: 'User1',
          },
          {
            path: '/user2',
            breadcrumbName: 'User2',
          },
        ],
      },
    ]}
  />
);
```

API

Breadcrumb

Property	Description	Type	Default	Version
itemRender	Custom item renderer	(route, params, routes, paths) => ReactNode	–	
params	Routing parameters	object	–	
items	The routing stack information of router	items[]	–	5.3.0
separator	Custom separator	ReactNode	/	

ItemType

type ItemType = RouteItemType | SeparatorType

RouteItemType

Property	Description	Type	Default	Version
className	The additional css class	string	–	
dropdownProps	The dropdown props	Dropdown	–	
href	Target of hyperlink. Can not work with `path`	string	–	
path	Connected path. Each path will connect with prev one. Can not work with `href`	string	–	
menu	The menu props	MenuProps	–	4.24.0
onClick	Set the handler to handle click event	(e:MouseEvent) => void	–	
title	item name	ReactNode	–	

SeparatorType

```
const item = {
  type: 'separator', // Must have
  separator: '/',
};
```

Property	Description	Type	Default	Version
type	Mark as separator	`separator`		5.3.0
separator	Custom separator	ReactNode	`/`	5.3.0

Use with browserHistory

The link of Breadcrumb item targets `#` by default, you can use `itemRender` to make a `browserHistory` Link.

```
import { Link } from 'react-router';

const items = [
  {
    path: 'index',
    title: 'home',
  },
  {
    path: 'first',
    title: 'first',
    children: [
      {
        path: '/general',
        title: 'General',
      },
      {
        path: '/layout',
        title: 'Layout',
```

```
      },
      {
        path: '/navigation',
        title: 'Navigation',
      },
    ],
  },
  {
    path: 'second',
    title: 'second',
  },
];
function itemRender(route, params, items, paths) {
  const last = items.indexOf(item) === items.length - 1;
  return last ? <span>{item.title}</span> : <Link to={paths.join('/')}>{item.title}</Link>;
}

return <Breadcrumb itemRender={itemRender} items={items} />;
```

Design Token

▼ Global Token

Token Name	Description	Type	Default Value
colorBgTextHover	Control the background color of text in hover state.	string	☐ rgba(0, 0, 0, 0.06)
colorPrimaryBorder	The stroke color under the main color gradient, used on the stroke of components such as Slider.	string	☐ #91caff
colorText	Default text color which comply with W3C standards, and this color is also the darkest neutral color.	string	☐ rgba(0, 0, 0, 0.88)
colorTextDescription	Control the font color of text description.	string	☐ rgba(0, 0, 0, 0.45)
borderRadiusSM	SM size border radius, used in small size components, such as Button, Input, Select and other input components in small size	number	4
fontFamily	The font family of Ant Design prioritizes the default interface font of the system, and provides a set of alternative font libraries that are suitable for screen display to maintain the readability and readability of the font under different platforms and browsers, reflecting the friendly, stable and professional characteristics.	string	-apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, 'Helvetica Neue', Arial, 'Noto Sans', sans-serif, 'Apple Color Emoji', 'Segoe UI Emoji', 'Segoe UI Symbol', 'Noto Color Emoji'
fontSize	The most widely used font size in the design system, from which the text gradient will be derived.	number	14
fontSizeIcon	Control the font size of operation icon in Select, Cascader, etc. Normally same as fontSizeSM.	number	12
lineHeight	Line height of text.	number	1.5714285714285714

Token Name	Description	Type	Default Value
lineWidthFocus	Control the width of the line when the component is in focus state.	`number`	4
marginXS	Control the margin of an element, with a small size.	`number`	8
marginXXS	Control the margin of an element, with the smallest size.	`number`	4
motionDurationMid	Motion speed, medium speed. Used for medium element animation interaction.	`string`	0.2s
paddingXXS	Control the extra extra small padding of the element.	`number`	4