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.

API

Breadcrumb

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

Breadcrumb.Item

Property	Description	Type	Default	Version
className	The additional css class	string	-	
dropdownProps	The dropdown props	[Dropdown](#)	-	
href	Target of hyperlink	string	-	
overlay	The dropdown menu	[Menu](#) | () => Menu	-	
onClick	Set the handler to handle click event	(e:MouseEvent) => void	-	

Breadcrumb.Separator

Property	Description	Type	Default	Version
children	Custom separator	ReactNode	/	

> When using `Breadcrumb.Separator` ,its parent component must be set to `separator=""` , otherwise the default separator of the parent component will appear.

routes

```
interface Route {
  path: string;
  breadcrumbName: string;
  children: Array<{
```

```
    path: string;
    breadcrumbName: string;
  }>;
}
```

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 routes = [
  {
    path: 'index',
    breadcrumbName: 'home',
  },
  {
    path: 'first',
    breadcrumbName: 'first',
    children: [
      {
        path: '/general',
        breadcrumbName: 'General',
      },
      {
        path: '/layout',
        breadcrumbName: 'Layout',
      },
      {
        path: '/navigation',
        breadcrumbName: 'Navigation',
      },
    ],
  },
  {
    path: 'second',
    breadcrumbName: 'second',
  },
];
function itemRender(route, params, routes, paths) {
  const last = routes.indexOf(route) === routes.length - 1;
  return last ? (
    <span>{route.breadcrumbName}</span>
  ) : (
    <Link to={paths.join('/')}>{route.breadcrumbName}</Link>
  );
}


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