

显示当前页面在系统层级结构中的位置，并能向上返回。

何时使用

- 当系统拥有超过两级以上的层级结构时；
- 当需要告知用户『你在哪里』时；
- 当需要向上导航的功能时。

API

Breadcrumb

参数	说明	类型	默认值	版本
itemRender	自定义链接函数，和 react-router 配置使用	(route, params, routes, paths) => ReactNode	-	
params	路由的参数	object	-	
routes	router 的路由栈信息	routes[]	-	
separator	分隔符自定义	ReactNode	/	

Breadcrumb.Item

参数	说明	类型	默认值	版本
className	自定义类名	string	-	
dropdownProps	弹出下拉菜单的自定义配置	Dropdown	-	
href	链接的目的地	string	-	
overlay	下拉菜单的内容	Menu () => Menu	-	
onClick	单击事件	(e:MouseEvent) => void	-	

Breadcrumb.Separator

参数	说明	类型	默认值	版本
children	要显示的分隔符	ReactNode	/	

注意：在使用 `Breadcrumb.Separator` 时，其父组件的分隔符必须设置为 `separator=""`，否则会出现父组件默认的分隔符。

routes

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

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

和 browserHistory 配合

和 react-router 一起使用时，默认生成的 url 路径是带有 # 的，如果和 browserHistory 一起使用的话，你可以使用 `itemRender` 属性定义面包屑链接。

```
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} />;
```