Handling the overall layout of a page.

Specification

Size

The first level navigation is left aligned near a logo, and the secondary menu is right aligned.

- Top Navigation: the height of the first level navigation `64px`, the second level navigation `48px`.
- Top Navigation (for landing pages): the height of the first level navigation `80px`, the second level navigation `56px`.
- Calculation formula of a top navigation: `48+8n`.
- Calculation formula of an aside navigation: `200+8n`.

Interaction rules

- The first level navigation and the last level navigation should be distinguishable by visualization;
- The current item should have the highest priority of visualization;
- When the current navigation item is collapsed, the style of the current navigation item is applied to its parent level;
- The left side navigation bar has support for both the accordion and expanding styles; you can choose the one that fits your case the best.

Visualization rules

Style of a navigation should conform to its level.

- **Emphasis by colorblock**

  When background color is a deep color, you can use this pattern for the parent level navigation item of the current page.

- **The highlight match stick**

  When background color is a light color, you can use this pattern for the current page navigation item; we recommend using it for the last item of the navigation path.

- **Highlighted font**

  From the visualization aspect, a highlighted font is stronger than colorblock; this pattern is often used for the parent level of the current item.

- **Enlarge the size of the font**

  `12px`, `14px` is a standard font size of navigations, `14px` is used for the first and the second level of the navigation. You can choose an appropriate font size regarding the level of your navigation.

Component Overview

- `Layout`: The layout wrapper, in which `Header` `Sider` `Content` `Footer` or `Layout` itself can be nested, and can be placed in any parent container.
- `Header`: The top layout with the default style, in which any element can be nested, and must be placed in `Layout`.

- `Sider` : The sidebar with default style and basic functions, in which any element can be nested, and must be placed in `Layout` .
- `Content` : The content layout with the default style, in which any element can be nested, and must be placed in `Layout` .
- `Footer` : The bottom layout with the default style, in which any element can be nested, and must be placed in `Layout` .

> *Based on `flex layout` , please pay attention to the compatibility.*

API

```
<Layout>
  <Header>header</Header>
  <Layout>
    <Sider>left sidebar</Sider>
    <Content>main content</Content>
    <Sider>right sidebar</Sider>
  </Layout>
  <Footer>footer</Footer>
</Layout>
```

Layout

The wrapper.

Property	Description	Type	Default
className	Container className	string	-
hasSider	Whether contain Sider in children, don't have to assign it normally. Useful in ssr avoid style flickering	boolean	-
style	To customize the styles	CSSProperties	-

Layout.Sider

The sidebar.

Property	Description	Type	Default
breakpoint	Breakpoints of the responsive layout	`xs` | `sm` | `md` | `lg` | `xl` | `xxl`	-
className	Container className	string	-
collapsed	To set the current status	boolean	-
collapsedWidth	Width of the collapsed sidebar, by setting to 0 a special trigger will appear	number	80
collapsible	Whether can be collapsed	boolean	false
defaultCollapsed	To set the initial status	boolean	false

reverseArrow	Reverse direction of arrow, for a sider that expands from the right	boolean	false
style	To customize the styles	CSSProperties	-
theme	Color theme of the sidebar	`light` | `dark`	`dark`
trigger	Specify the customized trigger, set to null to hide the trigger	ReactNode	-
width	Width of the sidebar	number | string	200
zeroWidthTriggerStyle	To customize the styles of the special trigger that appears when `collapsedWidth` is 0	object	-
onBreakpoint	The callback function, executed when breakpoints changed	(broken) => {}	-
onCollapse	The callback function, executed by clicking the trigger or activating the responsive layout	(collapsed, type) => {}	-

breakpoint width

```
{
  xs: '480px',
  sm: '576px',
  md: '768px',
  lg: '992px',
  xl: '1200px',
  xxl: '1600px',
}
```