When To Use

Navigation is an important part of any website, as a good navigation setup allows users to move around the site quickly and efficiently. Ant Design offers two navigation options: top and side. Top navigation provides all the categories and functions of the website. Side navigation provides the multi-level structure of the website.

More layouts with navigation: Layout.

Notes for developers

- Menu is rendered as a `ul` element, so it only supports `li` and `script-supporting elements` as children nodes。Your customized node should be wrapped by `Menu.Item`.
- Menu needs to collect its node structure, so its children should be `Menu.*` or encapsulated HOCs.

Examples

Top Navigation

```
import React, { useState } from 'react';
import { AppstoreOutlined, MailOutlined, SettingOutlined } from '@ant-design/icons';
import type { MenuProps } from 'antd';
import { Menu } from 'antd';

type MenuItem = Required<MenuProps>['items'][number];

const items: MenuItem[] = [
  {
    label: 'Navigation One',
    key: 'mail',
    icon: <MailOutlined />,
  },
  {
    label: 'Navigation Two',
    key: 'app',
    icon: <AppstoreOutlined />,
    disabled: true,
  },
  {
    label: 'Navigation Three - Submenu',
    key: 'SubMenu',
    icon: <SettingOutlined />,
    children: [
      {
        type: 'group',
        label: 'Item 1',
        children: [
          { label: 'Option 1', key: 'setting:1' },
          { label: 'Option 2', key: 'setting:2' },
        ],
      },
      {
        type: 'group',
        label: 'Item 2',
        children: [
          { label: 'Option 3', key: 'setting:3' },
          { label: 'Option 4', key: 'setting:4' },
        ],
      },
    ],
  },
  {
    key: 'alipay',
    label: (
      <a href="https://ant.design" target="_blank" rel="noopener
noreferrer">
        Navigation Four - Link
      </a>
    ),
  },
];

const App: React.FC = () => {
  const [current, setCurrent] = useState('mail');

  const onClick: MenuProps['onClick'] = (e) => {
    console.log('click ', e);
    setCurrent(e.key);
  };

  return <Menu onClick={onClick} selectedKeys={[current]} mode="horizontal"
items={items} />;
};

export default App;
```

Top Navigation (dark)

Debug

```
import React, { useState } from 'react';
import { AppstoreOutlined, MailOutlined, SettingOutlined } from '@ant-design/icons';
import type { MenuProps } from 'antd';
import { Menu } from 'antd';

type MenuItem = Required<MenuProps>['items'][number];

const items: MenuItem[] = [
  {
    label: 'Navigation One',
    key: 'mail',
    icon: <MailOutlined />,
  },
  {
    label: 'Navigation Two',
    key: 'app',
    icon: <AppstoreOutlined />,
    disabled: true,
  },
  {
    label: 'Navigation Three - Submenu',
    key: 'SubMenu',
    icon: <SettingOutlined />,
    children: [
      {
        type: 'group',
        label: 'Item 1',
        children: [
          { label: 'Option 1', key: 'setting:1' },
          { label: 'Option 2', key: 'setting:2' },
        ],
      },
      {
        type: 'group',
        label: 'Item 2',
        children: [
          { label: 'Option 3', key: 'setting:3' },
          { label: 'Option 4', key: 'setting:4' },
        ],
      },
    ],
  },
  {
    key: 'alipay',
    label: (
      <a href="https://ant.design" target="_blank" rel="noopener noreferrer">
        Navigation Four - Link
      </a>
    ),
  },
];

const App: React.FC = () => {
  const [current, setCurrent] = useState('mail');

  const onClick: MenuProps['onClick'] = (e) => {
    console.log('click ', e);
    setCurrent(e.key);
  };

  return (
    <Menu onClick={onClick} selectedKeys={[current]} mode="horizontal" items={items} theme="dark" />
  );
};

export default App;
```

Inline menu

```
import React from 'react';
import { AppstoreOutlined, MailOutlined, SettingOutlined } from '@ant-design/icons';
import type { MenuProps } from 'antd';
import { Menu } from 'antd';

type MenuItem = Required<MenuProps>['items'][number];

const items: MenuItem[] = [
  {
    key: 'sub1',
    label: 'Navigation One',
    icon: <MailOutlined />,
    children: [
      {
        key: 'g1',
        label: 'Item 1',
        type: 'group',
        children: [
          { key: '1', label: 'Option 1' },
          { key: '2', label: 'Option 2' },
        ],
      },
      {
        key: 'g2',
        label: 'Item 2',
        type: 'group',
        children: [
          { key: '3', label: 'Option 3' },
          { key: '4', label: 'Option 4' },
        ],
      },
    ],
  },
  {
    key: 'sub2',
    label: 'Navigation Two',
    icon: <AppstoreOutlined />,
    children: [
      { key: '5', label: 'Option 5' },
      { key: '6', label: 'Option 6' },
      {
        key: 'sub3',
        label: 'Submenu',
        children: [
          { key: '7', label: 'Option 7' },
          { key: '8', label: 'Option 8' },
        ],
      },
    ],
  },
  {
    type: 'divider',
  },
  {
    key: 'sub4',
    label: 'Navigation Three',
    icon: <SettingOutlined />,
    children: [
      { key: '9', label: 'Option 9' },
      { key: '10', label: 'Option 10' },
      { key: '11', label: 'Option 11' },
      { key: '12', label: 'Option 12' },
    ],
  },
  {
    key: 'grp',
    label: 'Group',
    type: 'group',
    children: [
      { key: '13', label: 'Option 13' },
      { key: '14', label: 'Option 14' },
    ],
  },
];

const App: React.FC = () => {
  const onClick: MenuProps['onClick'] = (e) => {
    console.log('click ', e);
  };

  return (
    <Menu
      onClick={onClick}
      style={{ width: 256 }}
      defaultSelectedKeys={['1']}
      defaultOpenKeys={['sub1']}
      mode="inline"
      items={items}
    />
  );
};

export default App;
```

Collapsed inline menu

```
import React, { useState } from 'react';
import {
  AppstoreOutlined,
  ContainerOutlined,
  DesktopOutlined,
  MailOutlined,
  MenuFoldOutlined,
  MenuUnfoldOutlined,
  PieChartOutlined,
} from '@ant-design/icons';
import type { MenuProps } from 'antd';
import { Button, Menu } from 'antd';

type MenuItem = Required<MenuProps>['items'][number];
```

```
const items: MenuItem[] = [
  { key: '1', icon: <PieChartOutlined />, label: 'Option 1' },
  { key: '2', icon: <DesktopOutlined />, label: 'Option 2' },
  { key: '3', icon: <ContainerOutlined />, label: 'Option 3' },
  {
    key: 'sub1',
    label: 'Navigation One',
    icon: <MailOutlined />,
    children: [
      { key: '5', label: 'Option 5' },
      { key: '6', label: 'Option 6' },
      { key: '7', label: 'Option 7' },
      { key: '8', label: 'Option 8' },
    ],
  },
  {
    key: 'sub2',
    label: 'Navigation Two',
    icon: <AppstoreOutlined />,
    children: [
      { key: '9', label: 'Option 9' },
      { key: '10', label: 'Option 10' },
      {
        key: 'sub3',
        label: 'Submenu',
        children: [
          { key: '11', label: 'Option 11' },
          { key: '12', label: 'Option 12' },
        ],
      },
    ],
  },
];

const App: React.FC = () => {
  const [collapsed, setCollapsed] = useState(false);

  const toggleCollapsed = () => {
    setCollapsed(!collapsed);
  };

  return (
    <div style={{ width: 256 }}>
      <Button type="primary" onClick={toggleCollapsed} style={{
marginBottom: 16 }}>
        {collapsed ? <MenuUnfoldOutlined /> : <MenuFoldOutlined />}
      </Button>
      <Menu
        defaultSelectedKeys={['1']}
        defaultOpenKeys={['sub1']}
        mode="inline"
        theme="dark"
        inlineCollapsed={collapsed}
        items={items}
      />
    </div>
  );
};

export default App;
```

Open current submenu only

```
import React, { useState } from 'react';
import { AppstoreOutlined, MailOutlined, SettingOutlined } from '@ant-
design/icons';
import type { MenuProps } from 'antd';
import { Menu } from 'antd';

type MenuItem = Required<MenuProps>['items'][number];

const items: MenuItem[] = [
  {
    key: '1',
    icon: <MailOutlined />,
    label: 'Navigation One',
    children: [
      { key: '11', label: 'Option 1' },
      { key: '12', label: 'Option 2' },
      { key: '13', label: 'Option 3' },
      { key: '14', label: 'Option 4' },
    ],
  },
  {
    key: '2',
    icon: <AppstoreOutlined />,
    label: 'Navigation Two',
    children: [
      { key: '21', label: 'Option 1' },
      { key: '22', label: 'Option 2' },
      {
        key: '23',
        label: 'Submenu',
        children: [
          { key: '231', label: 'Option 1' },
          { key: '232', label: 'Option 2' },
          { key: '233', label: 'Option 3' },
        ],
      },
      {
        key: '24',
        label: 'Submenu 2',
        children: [
          { key: '241', label: 'Option 1' },
          { key: '242', label: 'Option 2' },
          { key: '243', label: 'Option 3' },
        ],
      },
    ],
  },
  {
    key: '3',
    icon: <SettingOutlined />,
    label: 'Navigation Three',
    children: [
      { key: '31', label: 'Option 1' },
      { key: '32', label: 'Option 2' },
      { key: '33', label: 'Option 3' },
      { key: '34', label: 'Option 4' },
    ],
  },
];

interface LevelKeysProps {
  key?: string;
  children?: LevelKeysProps[];
}

const getLevelKeys = (items1: LevelKeysProps[]) => {
  const key: Record<string, number> = {};
  const func = (items2: LevelKeysProps[], level = 1) => {
    items2.forEach((item) => {
      if (item.key) {
        key[item.key] = level;
      }
      if (item.children) {
        func(item.children, level + 1);
      }
    });
  };
  func(items1);
  return key;
};

const levelKeys = getLevelKeys(items as LevelKeysProps[]);

const App: React.FC = () => {
  const [stateOpenKeys, setStateOpenKeys] = useState(['2', '23']);

  const onOpenChange: MenuProps['onOpenChange'] = (openKeys) => {
    const currentOpenKey = openKeys.find((key) =>
stateOpenKeys.indexOf(key) === -1);
    // open
    if (currentOpenKey !== undefined) {
      const repeatIndex = openKeys
        .filter((key) => key !== currentOpenKey)
        .findIndex((key) => levelKeys[key] === levelKeys[currentOpenKey]);

      setStateOpenKeys(
        openKeys
          // remove repeat key
          .filter((_, index) => index !== repeatIndex)
          // remove current level all child
          .filter((key) => levelKeys[key] <= levelKeys[currentOpenKey]),
      );
    } else {
      // close
      setStateOpenKeys(openKeys);
    }
  };

  return (
    <Menu
      mode="inline"
      defaultSelectedKeys={['231']}
      openKeys={stateOpenKeys}
      onOpenChange={onOpenChange}
      style={{ width: 256 }}
      items={items}
    />
  );
};

export default App;
```

Vertical menu

```
import React from 'react';
import { AppstoreOutlined, MailOutlined, SettingOutlined } from '@ant-design/icons';
import type { MenuProps } from 'antd';
import { Menu } from 'antd';

type MenuItem = Required<MenuProps>['items'][number];

const items: MenuItem[] = [
  {
    key: 'sub1',
    icon: <MailOutlined />,
    label: 'Navigation One',
    children: [
      {
        key: '1-1',
        label: 'Item 1',
        type: 'group',
        children: [
          { key: '1', label: 'Option 1' },
          { key: '2', label: 'Option 2' },
        ],
      },
      {
        key: '1-2',
        label: 'Item 2',
        type: 'group',
        children: [
          { key: '3', label: 'Option 3' },
          { key: '4', label: 'Option 4' },
        ],
      },
    ],
  },
  {
    key: 'sub2',
    icon: <AppstoreOutlined />,
    label: 'Navigation Two',
    children: [
      { key: '5', label: 'Option 5' },
      { key: '6', label: 'Option 6' },
      {
        key: 'sub3',
        label: 'Submenu',
        children: [
          { key: '7', label: 'Option 7' },
          { key: '8', label: 'Option 8' },
        ],
      },
    ],
  },
  {
    key: 'sub4',
    label: 'Navigation Three',
    icon: <SettingOutlined />,
    children: [
      { key: '9', label: 'Option 9' },
      { key: '10', label: 'Option 10' },
      { key: '11', label: 'Option 11' },
      { key: '12', label: 'Option 12' },
    ],
  },
];

const onClick: MenuProps['onClick'] = (e) => {
  console.log('click', e);
};

const App: React.FC = () => (
  <Menu onClick={onClick} style={{ width: 256 }} mode="vertical" items=
{items} />
);

export default App;
```

Menu Themes

```
import React, { useState } from 'react';
import { AppstoreOutlined, MailOutlined, SettingOutlined } from '@ant-
design/icons';
import type { MenuProps, MenuTheme } from 'antd';
import { Menu, Switch } from 'antd';

type MenuItem = Required<MenuProps>['items'][number];

const items: MenuItem[] = [
  {
    key: 'sub1',
    label: 'Navigation One',
    icon: <MailOutlined />,
    children: [
      { key: '1', label: 'Option 1' },
      { key: '2', label: 'Option 2' },
      { key: '3', label: 'Option 3' },
      { key: '4', label: 'Option 4' },
    ],
  },
  {
    key: 'sub2',
    label: 'Navigation Two',
    icon: <AppstoreOutlined />,
    children: [
      { key: '5', label: 'Option 5' },
      { key: '6', label: 'Option 6' },
      {
        key: 'sub3',
        label: 'Submenu',
        children: [
          { key: '7', label: 'Option 7' },
          { key: '8', label: 'Option 8' },
        ],
      },
    ],
  },
  {
    key: 'sub4',
    label: 'Navigation Three',
    icon: <SettingOutlined />,
    children: [
      { key: '9', label: 'Option 9' },
      { key: '10', label: 'Option 10' },
      { key: '11', label: 'Option 11' },
      { key: '12', label: 'Option 12' },
    ],
  },
];

const App: React.FC = () => {
  const [theme, setTheme] = useState<MenuTheme>('dark');
  const [current, setCurrent] = useState('1');

  const changeTheme = (value: boolean) => {
    setTheme(value ? 'dark' : 'light');
  };

  const onClick: MenuProps['onClick'] = (e) => {
    console.log('click ', e);
    setCurrent(e.key);
  };

  return (
    <>
      <Switch
        checked={theme === 'dark'}
        onChange={changeTheme}
        checkedChildren="Dark"
        unCheckedChildren="Light"
      />
      <br />
      <br />
      <Menu
        theme={theme}
        onClick={onClick}
        style={{ width: 256 }}
        defaultOpenKeys={['sub1']}
        selectedKeys={[current]}
        mode="inline"
        items={items}
      />
    </>
  );
};

export default App;
```

Sub-menu theme

```
import React, { useState } from 'react';
import { MailOutlined } from '@ant-design/icons';
import type { MenuProps, MenuTheme } from 'antd';
import { Menu, Switch } from 'antd';

type MenuItem = Required<MenuProps>['items'][number];

const App: React.FC = () => {
  const [menuTheme, setMenuTheme] = useState<MenuTheme>('light');
  const [current, setCurrent] = useState('1');

  const changeTheme = (value: boolean) => {
    setMenuTheme(value ? 'dark' : 'light');
  };
  const onClick: MenuProps['onClick'] = (e) => {
    setCurrent(e.key);
  };

  const items: MenuItem[] = [
    {
      key: 'sub1',
      icon: <MailOutlined />,
      label: 'Navigation One',
      theme: menuTheme,
      children: [
        { key: '1', label: 'Option 1' },
        { key: '2', label: 'Option 2' },
        { key: '3', label: 'Option 3' },
      ],
    },
    { key: '5', label: 'Option 5' },
    { key: '6', label: 'Option 6' },
  ];

  return (
    <>
      <Switch
        checked={menuTheme === 'dark'}
        onChange={changeTheme}
        checkedChildren="Dark"
        unCheckedChildren="Light"
      />
      <br />
      <br />
      <Menu
        onClick={onClick}
        style={{ width: 256 }}
        openKeys={['sub1']}
        selectedKeys={[current]}
        mode="vertical"
        theme="dark"
        items={items}
        getPopupContainer={(node) => node.parentNode as HTMLElement}
      />
    </>
  );
};

export default App;
```

Switch the menu type

```
import React, { useState } from 'react';
import {
  AppstoreOutlined,
  CalendarOutlined,
  LinkOutlined,
  MailOutlined,
  SettingOutlined,
} from '@ant-design/icons';
import { Divider, Menu, Switch } from 'antd';
import type { GetProp, MenuProps } from 'antd';

type MenuTheme = GetProp<MenuProps, 'theme'>;

type MenuItem = GetProp<MenuProps, 'items'>[number];

const items: MenuItem[] = [
  {
    key: '1',
    icon: <MailOutlined />,
    label: 'Navigation One',
  },
  {
    key: '2',
    icon: <CalendarOutlined />,
    label: 'Navigation Two',
  },
  {
    key: 'sub1',
    label: 'Navigation Two',
    icon: <AppstoreOutlined />,
    children: [
      { key: '3', label: 'Option 3' },
      { key: '4', label: 'Option 4' },
      {
        key: 'sub1-2',
        label: 'Submenu',
        children: [
          { key: '5', label: 'Option 5' },
          { key: '6', label: 'Option 6' },
        ],
      },
    ],
  },
  {
    key: 'sub2',
    label: 'Navigation Three',
    icon: <SettingOutlined />,
    children: [
      { key: '7', label: 'Option 7' },
      { key: '8', label: 'Option 8' },
      { key: '9', label: 'Option 9' },
      { key: '10', label: 'Option 10' },
    ],
  },
  {
    key: 'link',
    icon: <LinkOutlined />,
    label: (
      <a href="https://ant.design" target="_blank" rel="noopener noreferrer">
        Ant Design
      </a>
    ),
  },
];

const App: React.FC = () => {
  const [mode, setMode] = useState<'vertical' | 'inline'>('inline');
  const [theme, setTheme] = useState<MenuTheme>('light');

  const changeMode = (value: boolean) => {
    setMode(value ? 'vertical' : 'inline');
  };

  const changeTheme = (value: boolean) => {
    setTheme(value ? 'dark' : 'light');
  };

  return (
    <>
      <Switch onChange={changeMode} /> Change Mode
      <Divider type="vertical" />
      <Switch onChange={changeTheme} /> Change Style
      <br />
      <br />
      <Menu
        style={{ width: 256 }}
        defaultSelectedKeys={['1']}
        defaultOpenKeys={['sub1']}
        mode={mode}
        theme={theme}
        items={items}
      />
    </>
  );
};

export default App;
```

Style debug

Debug

```
import React, { useState } from 'react';
import { AppstoreOutlined, MailOutlined } from '@ant-design/icons';
import type { MenuProps, MenuTheme } from 'antd';
import { Menu, Switch } from 'antd';

type MenuItem = Required<MenuProps>['items'][number];

const items: MenuItem[] = [
  {
    key: 'sub1',
    label: 'Navigation One Long Long Long Long',
    icon: <MailOutlined />,
    children: [
      { key: '1', label: 'Option 1' },
      { key: '2', label: 'Option 2' },
      { key: '3', label: 'Option 3' },
      { key: '4', label: 'Option 4' },
    ],
  },
  {
    key: 'sub2',
    label: 'Navigation Two',
    icon: <AppstoreOutlined />,
    children: [
      { key: '5', label: 'Option 5' },
      { key: '6', label: 'Option 6' },
      {
        key: 'sub3',
        label: 'Submenu',
        children: [
          { key: '7', label: 'Option 7' },
          { key: '8', label: 'Option 8' },
        ],
      },
    ],
  },
  { key: '11', label: 'Option 11' },
  { key: '12', label: 'Option 12' },
];

const App: React.FC = () => {
  const [menuTheme, setMenuTheme] = useState<MenuTheme>('dark');
  const [current, setCurrent] = useState('1');

  const changeTheme = (value: boolean) => {
    setMenuTheme(value ? 'dark' : 'light');
  };

  const onClick: MenuProps['onClick'] = (e) => {
    console.log('click ', e);
    setCurrent(e.key);
  };

  return (
    <>
      <Switch
        checked={menuTheme === 'dark'}
        onChange={changeTheme}
        checkedChildren="Dark"
        unCheckedChildren="Light"
      />
      <br />
      <br />
      <Menu
        theme={menuTheme}
        onClick={onClick}
        selectedKeys={[current]}
        mode="inline"
        items={items}
        inlineCollapsed
        // Test only. Remove in future.
        _internalRenderMenuItem={(node) =>
          React.cloneElement<any>(node, {
            style: {
              ...(node as any).props.style,
              textDecoration: 'underline',
            },
          })
        }
        // Test only. Remove in future.
        _internalRenderSubMenuItem={(node) =>
          React.cloneElement<any>(node, {
            style: {
              ...(node as any).props.style,
              background: 'rgba(255, 255, 255, 0.3)',
            },
          })
        }
        // Test only. Remove in future.
        _internalDisableMenuItemTitleTooltip
      />
    </>
  );
};

export default App;
```

Menu v4

Debug

```
import React, { useState } from 'react';
import {
  AppstoreOutlined,
  CalendarOutlined,
  LinkOutlined,
  MailOutlined,
  SettingOutlined,
} from '@ant-design/icons';
import { ConfigProvider, Menu, Switch, Typography } from 'antd';
import type { MenuProps } from 'antd';

type MenuItem = Required<MenuProps>['items'][number];

const items: MenuItem[] = [
  {
    key: '1',
    icon: <MailOutlined />,
    label: 'Navigation One',
  },
  {
    key: '2',
    icon: <CalendarOutlined />,
    label: 'Navigation Two',
  },
  {
    key: 'sub1',
    icon: <AppstoreOutlined />,
    label: 'Navigation Two',
    children: [
      {
        key: '3',
        label: (
          <Typography.Text ellipsis>
            Ant Design, a design language for background applications, is refined by Ant UED Team
          </Typography.Text>
        ),
      },
      {
        key: '4',
        label: 'Option 4',
      },
      {
        key: 'sub1-2',
        label: 'Submenu',
        children: [
          { key: '5', label: 'Option 5' },
          { key: '6', label: 'Option 6' },
        ],
      },
    ],
  },
  {
    key: 'sub2',
    label: 'Navigation Three',
    icon: <SettingOutlined />,
    children: [
      { label: 'Option 7', key: '7' },
      { label: 'Option 8', key: '8' },
      { label: 'Option 9', key: '9' },
      { label: 'Option 10', key: '10' },
    ],
  },
  {
    key: 'link',
    icon: <LinkOutlined />,
    label: (
      <a href="https://ant.design" target="_blank" rel="noopener noreferrer">
        Ant Design
      </a>
    ),
  },
];

const App: React.FC = () => {
  const [mode, setMode] = useState<'vertical' | 'inline'>('inline');

  const changeMode = (value: boolean) => {
    setMode(value ? 'vertical' : 'inline');
  };

  return (
    <>
      <Switch onChange={changeMode} /> Change Mode
      <br />
      <br />
      <ConfigProvider
        theme={{
          components: {
            Menu: {
              itemBorderRadius: 0,
              subMenuItemBorderRadius: 0,
              itemHoverColor: '#1890ff',
              itemSelectedColor: '#1890ff',
              itemSelectedBg: '#e6f7ff',
              activeBarWidth: 3,
              itemMarginInline: 0,
              itemHoverBg: 'transparent',
            },
          },
        }}
      >
        <Menu
          style={{ width: 256 }}
          defaultSelectedKeys={['1']}
          defaultOpenKeys={['sub1']}
          mode={mode}
          items={items}
        />
      </ConfigProvider>
    </>
  );
};

export default App;
```

Component Token

Debug

```
import React, { useState } from 'react';
import {
  AppstoreOutlined,
  ContainerOutlined,
  DesktopOutlined,
  MailOutlined,
  PieChartOutlined,
  SettingOutlined,
} from '@ant-design/icons';
import type { MenuProps } from 'antd';
import { ConfigProvider, Menu, Space, theme } from 'antd';

type MenuItem = Required<MenuProps>['items'][number];

const items: MenuItem[] = [
  {
    label: 'Navigation One',
    key: 'mail',
    icon: <MailOutlined />,
  },
  {
    label: 'Navigation Two',
    key: 'app',
    icon: <AppstoreOutlined />,
    disabled: true,
  },
  {
    label: 'Navigation Three - Submenu',
    key: 'SubMenu',
    icon: <SettingOutlined />,
    children: [
      {
        type: 'group',
        label: 'Item 1',
        children: [
          { label: 'Option 1', key: 'setting:1' },
          { label: 'Option 2', key: 'setting:2' },
        ],
      },
      {
        type: 'group',
        label: 'Item 2',
        children: [
          { label: 'Option 3', key: 'setting:3' },
          { label: 'Option 4', key: 'setting:4' },
        ],
      },
    ],
  },
  {
    key: 'alipay',
    label: (
      <a href="https://ant.design" target="_blank" rel="noopener
noreferrer">
        Navigation Four - Link
      </a>
    ),
  },
];

const items2: MenuItem[] = [
  {
    key: '1',
    icon: <PieChartOutlined />,
    label: 'Option 1',
  },
  {
    key: '2',
    icon: <DesktopOutlined />,
    label: 'Option 2',
  },
  {
    key: '3',
    icon: <ContainerOutlined />,
    label: 'Option 3',
  },
  {
    key: 'sub1',
    label: 'Navigation One',
    icon: <MailOutlined />,
    children: [
      { key: '5', label: 'Option 5' },
      { key: '6', label: 'Option 6' },
      { key: '7', label: 'Option 7' },
      { key: '8', label: 'Option 8' },
    ],
  },
  {
    key: 'sub2',
    label: 'Navigation Two',
    icon: <AppstoreOutlined />,
    children: [
      { key: '9', label: 'Option 9' },
      { key: '10', label: 'Option 10' },
      {
        key: 'sub3',
        label: 'Submenu',
        children: [
          { key: '11', label: 'Option 11' },
          { key: '12', label: 'Option 12' },
        ],
      },
    ],
  },
];

const App: React.FC = () => {
  const [current, setCurrent] = useState('mail');

  const onClick: MenuProps['onClick'] = (e) => {
    console.log('click ', e);
    setCurrent(e.key);
  };

  return (
    <Space direction="vertical">
      <ConfigProvider
        theme={{
          algorithm: [theme.darkAlgorithm],
          components: {
            Menu: {
              popupBg: 'yellow',
              darkPopupBg: 'red',
            },
          },
        }}
      >
        <Menu onClick={onClick} selectedKeys={[current]} mode="horizontal"
items={items} />
        <Menu
          defaultSelectedKeys={['1']}
          defaultOpenKeys={['sub1']}
          mode="inline"
          theme="dark"
          inlineCollapsed
          items={items2}
          style={{
            width: 56,
          }}
        />
      </ConfigProvider>
      <ConfigProvider
        theme={{
          components: {
            Menu: {
              horizontalItemBorderRadius: 6,
              popupBg: 'red',
              horizontalItemHoverBg: '#f5f5f5',
            },
          },
        }}
      >
        <Menu onClick={onClick} selectedKeys={[current]} mode="horizontal" items={items} />
      </ConfigProvider>
      <ConfigProvider
        theme={{
          components: {
            Menu: {
              darkItemColor: '#91daff',
              darkItemBg: '#d48806',
              darkSubMenuItemBg: '#faad14',
              darkItemSelectedColor: '#ffccc7',
              darkItemSelectedBg: '#52c41a',
            },
          },
        }}
      >
        <Menu
          defaultSelectedKeys={['1']}
          defaultOpenKeys={['sub1']}
          mode="inline"
          theme="dark"
          items={items2}
          style={{
            width: 256,
          }}
        />
      </ConfigProvider>
    </Space>
```

```
  );
};

export default App;
```

Extra Style debug

Debug

```
import React from 'react';
import { DownOutlined, MailOutlined } from '@ant-design/icons';
import type { MenuProps } from 'antd';
import { Flex, Menu, Space } from 'antd';

type MenuItem = Required<MenuProps>['items'][number];

const items1: MenuItem[] = [
  {
    key: 'sub1',
    icon: <MailOutlined />,
    label: 'Navigation One',
    children: [
      {
        key: '1',
        label: (
          <Flex justify="space-between">
            <span>Option 1</span>
            <DownOutlined />
          </Flex>
        ),
      },
      {
        key: '2',
        label: 'Option 2',
        extra: '⌘P',
      },
      {
        key: '3',
        label: <a href="https://www.baidu.com">Link Option</a>,
        disabled: true,
      },
    ],
  },
];

const items2: MenuItem[] = [
  { key: '1', label: 'Users', extra: '⌘U' },
  { key: '2', label: 'Profile', extra: '⌘P' },
];

const App: React.FC = () => (
  <Space direction="vertical">
    <Menu
      mode="inline"
      defaultOpenKeys={['sub1']}
      defaultSelectedKeys={['1']}
      style={{ width: 256 }}
      items={items1}
    />
    <Menu theme="dark" style={{ width: 256 }} items={items2} />
  </Space>
);

export default App;
```

API

Common props ref: Common props

Menu

Param	Description	Type	Default value	Versic
defaultOpenKeys	Array with the keys of default opened sub menus	string[]	-	
defaultSelectedKeys	Array with the keys of default selected menu items	string[]	-	
expandIcon	custom expand icon of submenu	ReactNode | `(props: SubMenuProps & { isSubMenu: boolean }) => ReactNode`	-	4.9.0
forceSubMenuRender	Render submenu	boolean	false	

	into DOM before it becomes visible			
inlineCollapsed	Specifies the collapsed status when menu is inline mode	boolean	-	
inlineIndent	Indent (in pixels) of inline menu items on each level	number	24	
items	Menu item content	[ItemType[]]()	-	4.20.0
mode	Type of menu	`vertical` | `horizontal` | `inline`	`vertical`	
multiple	Allows selection of multiple items	boolean	false	
openKeys	Array with the keys of currently opened sub-menus	string[]	-	
overflowedIndicator	Customized the ellipsis icon when menu is collapsed horizontally	ReactNode	`<EllipsisOutlined />`	
selectable	Allows selecting menu items	boolean	true	

selectedKeys	Array with the keys of currently selected menu items	string[]	-	
style	Style of the root node	CSSProperties	-	
subMenuCloseDelay	Delay time to hide submenu when mouse leaves (in seconds)	number	0.1	
subMenuOpenDelay	Delay time to show submenu when mouse enters, (in seconds)	number	0	
theme	Color theme of the menu	`light` | `dark`	`light`	
triggerSubMenuAction	Which action can trigger submenu open/close	`hover` | `click`	`hover`	
onClick	Called when a menu item is clicked	function({ item, key, keyPath, domEvent })	-	
onDeselect	Called when a menu item is deselected (multiple mode only)	function({ item, key, keyPath, selectedKeys, domEvent })	-	

onOpenChange	Called when sub-menus are opened or closed	function(openKeys: string[])	-	
onSelect	Called when a menu item is selected	function({ item, key, keyPath, selectedKeys, domEvent })	-	

> *More options in rc-menu*

ItemType

> *type ItemType = MenuItemType | SubMenuType | MenuItemGroupType | MenuDividerType;*

MenuItemType

Param	Description	Type	Default value	Version
danger	Display the danger style	boolean	false	
disabled	Whether menu item is disabled	boolean	false	
extra	The extra of the menu item	ReactNode	-	5.21.0
icon	The icon of the menu item	ReactNode	-	
key	Unique ID of the menu item	string	-	
label	Menu label	ReactNode	-	
title	Set display title for collapsed item	string	-	

SubMenuType

Property	Description	Type	Default value	Version
children	Sub-menus or sub-menu items	ItemType[]	-	
disabled	Whether sub-menu is disabled	boolean	false	
icon	Icon of sub menu	ReactNode	-	
key	Unique ID of the sub-menu	string	-	
label	Menu label	ReactNode	-	

popupClassName	Sub-menu class name, not working when `mode="inline"`	string	-	
popupOffset	Sub-menu offset, not working when `mode="inline"`	[number, number]	-	
theme	Color theme of the SubMenu (inherits from Menu by default)		`light` | `dark`	-
onTitleClick	Callback executed when the sub-menu title is clicked	function({ key, domEvent })	-	

MenuItemGroupType

Define `type` as `group` to make as group:

```
const groupItem = {
  type: 'group', // Must have
  label: 'My Group',
  children: [],
};
```

Param	Description	Type	Default value	Version
children	Sub-menu items	MenuItemType[]	-	
label	The title of the group	ReactNode	-	

MenuDividerType

Divider line in between menu items, only used in vertical popup Menu or Dropdown Menu. Need define the `type` as `divider`：

```
const dividerItem = {
  type: 'divider', // Must have
};
```

Param	Description	Type	Default value	Version
dashed	Whether line is dashed	boolean	false	

FAQ

Why will Menu's children be rendered twice?

Menu collects structure info with twice-render to support HOC usage. Merging into one render may cause the logic to become much more complex. Contributions to help improve the collection logic are welcomed.

Why Menu do not responsive collapse in Flex layout?

Menu will render fully item in flex layout and then collapse it. You need tell flex not consider Menu width to enable responsive (online demo):

```
<div style={{ flex }}>
  <div style={{ ... }}>Some Content</div>
  <Menu style={{ minWidth: 0, flex: "auto" }} />
</div>
```

Design Token