

Tour

A popup component for guiding users through a product. Available since `5.0.0` .

When To Use

Use when you want to guide users through a product.

Examples

Begin Tour

Upload

Save

...

Basic

The most basic usage.

```
import React, { useRef, useState } from 'react'
import { Button, Divider, Space, Tour } from '@ant-design/compatible'
import type { TourProps } from 'antd';
import { EllipsisOutlined } from '@ant-design/icons';

const App: React.FC = () => {
  const ref1 = useRef(null);
  const ref2 = useRef(null);
  const ref3 = useRef(null);

  const [open, setOpen] = useState<boolean>(false)

  const steps: TourProps['steps'] = [
    {
      title: 'Upload File',
      description: 'Put your files here.',
      cover: (
         />
      ),
      target: () => ref1.current,
    },
    {
      title: 'Save',
      description: 'Save your changes.',
      target: () => ref2.current,
    },
    {
      title: 'Other Actions',
      description: 'Click to see other actions',
      target: () => ref3.current,
    },
  ],

  return (
    <>
      <Button type="primary" onClick={() => setBeginTour(true)}>
        Begin Tour
      </Button>

      <Divider />

      <Space>
        <Button ref={ref1}> Upload</Button>
        <Button ref={ref2} type="primary"> Save
          </Button>
        <Button ref={ref3} icon={EllipsisOutlined} type="primary"> More
          </Button>
      </Space>

      <Tour open={open} onClose={() => setOpen(false)} steps={steps}>
      </>
    </>
  );
};

export default App;
```

Begin non-modal Tour

Upload

Save

...

Non-modal

Use `mask={false}` to make Tour non-modal. At

the meantime it is recommended to use with

`type="primary"` to emphasize the guide itself.

```
import React, { useRef, useState } from 'react'
import { Button, Divider, Space, Tour } from 'antd'
import type { TourProps } from 'antd'
import { EllipsisOutlined } from '@ant-design/icons'

const App: React.FC = () => {
  const ref1 = useRef(null);
  const ref2 = useRef(null);
  const ref3 = useRef(null);

  const [open, setOpen] = useState<boolean>(false);

  const steps: TourProps['steps'] = [
    {
      title: 'Upload File',
      description: 'Put your files here.',
      cover: (
        
      ),
      target: () => ref1.current,
    },
    {
      title: 'Save',
      description: 'Save your changes.',
      target: () => ref2.current,
    },
    {
      title: 'Other Actions',
      description: 'Click to see other actions',
      target: () => ref3.current,
    },
  ];

  return (
    <>
      <Button type="primary" onClick={() => setOpen(true)}>
        Begin non-modal Tour
      </Button>

      <Divider />

      <Space>
        <Button ref={ref1}> Upload</Button>
        <Button ref={ref2} type="primary">
          Save
        </Button>
        <Button ref={ref3} icon={EllipsisOutlined} />
      </Space>

      <Tour open={open} onClose={() => setOpen(false)} steps={steps} />
    </>
  );
};
```

Begin Tour

Placement [↗](#)

Change the placement of the guide relative to the target, there are 12 placements available. When

`target={null}` the guide will show in the center.

```
import React, { useRef, useState } from 'react'
import { Button, Tour } from 'antd';
import type { TourProps } from 'antd';

const App: React.FC = () => {
  const ref = useRef(null);

  const [open, setOpen] = useState<boolean>(false);

  const steps: TourProps['steps'] = [
    {
      title: 'Center',
      description: 'Displayed in the center of target: null',
    },
    {
      title: 'Right',
      description: 'On the right of target.',
      placement: 'right',
      target: () => ref.current,
    },
    {
      title: 'Top',
      description: 'On the top of target.',
      placement: 'top',
      target: () => ref.current,
    },
  ];

  return (
    <>
      <Button type="primary" onClick={() => setOpen(true)}>
        Begin Tour
      </Button>

      <Tour open={open} onClose={() => setOpen(false)}>
        {steps}
      </Tour>
    </>
  );
};

export default App;
```

Begin Tour

Upload

Save

...

custom mask style [↗](#)

custom mask style.

```
import React, { useRef, useState } from 'react'
import { Button, Divider, Space, Tour } from 'antd';
import type { TourProps } from 'antd';
import { EllipsisOutlined } from '@ant-design/icons';

const App: React.FC = () => {
  const ref1 = useRef(null);
  const ref2 = useRef(null);
  const ref3 = useRef(null);

  const [open, setOpen] = useState<boolean>(false);

  const steps: TourProps['steps'] = [
    {
      title: 'Upload File',
      description: 'Put your files here.',
      cover: (
        
      ),
      target: () => ref1.current,
    },
    {
      title: 'Save',
      description: 'Save your changes.',
      target: () => ref2.current,
      mask: {
        style: {
          boxShadow: 'inset 0 0 15px #fff',
        },
        color: 'rgba(40, 0, 255, .4)',
      },
    },
    {
      title: 'Other Actions',
      description: 'Click to see other actions',
      target: () => ref3.current,
      mask: false,
    },
  ];

  return (
    <>
      <Button type="primary" onClick={() => setOpen(true)}>
        Begin Tour
      </Button>

      <Divider />

      <Space>
        <Button ref={ref1}> Upload</Button>
        <Button ref={ref2} type="primary">
          Save
        </Button>
        <Button ref={ref3} icon={EllipsisOutlined} />
      </Space>

      <Tour

```

Begin Tour

Upload

Save

...

custom indicator [✎](#)

Custom indicator.

```
import { EllipsisOutlined } from '@ant-design/
import type { TourProps } from 'antd';
import { Button, Divider, Space, Tour } from '
import React, { useRef, useState } from 'react
```

```
const App: React.FC = () => {
  const ref1 = useRef<HTMLButtonElement>(null)
  const ref2 = useRef<HTMLButtonElement>(null)
  const ref3 = useRef<HTMLButtonElement>(null)
```

```
  const [open, setOpen] = useState<boolean>(false)
```

```
  const steps: TourProps['steps'] = [
    {
      title: 'Upload File',
      description: 'Put your files here.',
      target: () => ref1.current!,
    },
    {
      title: 'Save',
      description: 'Save your changes.',
      target: () => ref2.current!,
    },
    {
      title: 'Other Actions',
      description: 'Click to see other actions',
      target: () => ref3.current!,
    },
  ],
];
```

```
  return (
    <>
      <Button type="primary" onClick={() => setOpen(true)}>
        Begin Tour
      </Button>
      <Divider />
      <Space>
        <Button ref={ref1}>Upload</Button>
        <Button ref={ref2} type="primary">
          Save
        </Button>
        <Button ref={ref3} icon={<EllipsisOutlined />} />
      </Space>
      <Tour
        open={open}
        onClose={() => setOpen(false)}
        steps={steps}
        indicatorsRender={(current, total) => (
          <span>
            {current + 1} / {total}
          </span>
        )}
      />
    </>
  );
};
```

```
export default App;
```

```
open={open}
onClose={() => setOpen(false)}
steps={steps}
mask={{
  style: {
    boxShadow: 'inset 0 0 15px #333',
  },
  color: 'rgba(80, 255, 255, .4)',
}}
/>
</>
);
};
```

```
export default App;
```

API

Tour

Property	Description	Type	Default	Version
arrow	Whether to show the arrow, including the configuration whether to point to the center of the element	<code>boolean { pointAtCenter: boolean }</code>	<code>true</code>	
placement	Position of the guide card relative to the target element	<code>left leftTop leftBottom right rightTop rightBottom top topLeft topRight bottom bottomLeft bottomRight</code>	<code>bottom</code>	
onClose	Callback function on shutdown	<code>Function</code>	<code>-</code>	
mask	Whether to enable masking, change mask style and fill color by pass custom props	<code>boolean { style?: React.CSSProperties; color?: string; }</code>	<code>true</code>	
type	Type, affects the background color and text color	<code>default primary</code>	<code>default</code>	
open	Open tour	<code>boolean</code>	<code>-</code>	
onChange	Callback when the step changes. Current is the previous step	<code>(current: number) => void</code>	<code>-</code>	
current	What is the current step	<code>number</code>	<code>-</code>	
scrollIntoViewOptions	support pass custom scrollIntoView options	<code>boolean ScrollIntoViewOptions</code>	<code>true</code>	5.0.0
indicatorsRender	custom indicator	<code>(current: number, total: number) => ReactNode</code>	<code>-</code>	5.0.0
zIndex	Tour's zIndex	<code>number</code>	<code>1001</code>	5.0.0

TourStep

Property	Description	Type	Default	Version
target	Get the element the guide card points	<code>() => HTMLElement</code> <code>HTMLElement</code>	<code>-</code>	

Property	Description	Type	Default	Version
	to. Empty makes it show in center of screen			
arrow	Whether to show the arrow, including the configuration whether to point to the center of the element	<code>boolean</code> { <code>pointAtCenter:</code> <code>boolean</code> }	<code>true</code>	
cover	Displayed pictures or videos	<code>ReactNode</code>	-	
title	title	<code>ReactNode</code>	-	
description	description	<code>ReactNode</code>	-	
placement	Position of the guide card relative to the target element	<code>left</code> <code>leftTop</code> <code>leftBottom</code> <code>right</code> <code>rightTop</code> <code>rightBottom</code> <code>top</code> <code>topLeft</code> <code>topRight</code> <code>bottom</code> <code>bottomLeft</code> <code>bottomRight</code>	<code>bottom</code>	
onClose	Callback function on shutdown	<code>Function</code>	-	
mask	Whether to enable masking, change mask style and fill color by pass custom props, the default follows the <code>mask</code> property of Tour	<code>boolean</code> { <code>style?:</code> <code>React.CSSProperties;</code> <code>color?:</code> <code>string;</code> }	<code>true</code>	
type	Type, affects the background color and text color	<code>default</code> <code>primary</code>	<code>default</code>	
nextButtonProps	Properties of the Next button	{ <code>children:</code> <code>ReactNode;</code> <code>onClick:</code> <code>Function</code> }	-	
prevButtonProps	Properties of the previous button	{ <code>children:</code> <code>ReactNode;</code> <code>onClick:</code> <code>Function</code> }	-	
scrollIntoViewOptions	support pass custom <code>scrollIntoView</code> options, the default follows the <code>scrollIntoViewOptions</code> property of Tour	<code>boolean</code> <code>ScrollIntoViewOptions</code>	<code>true</code>	5

Design Token

▼ Global Token

Token Name	Description	Type	Default Value
colorBgContainer	Container background color, e.g: default button, input box, etc. Be sure not to	<code>string</code>	<code>#ffffff</code>

Token Name	Description	Type	Default Value
	confuse this with `colorBgElevated`.		
colorBgTextHover	Control the background color of text in hover state.	string	<code>rgba(0, 0, 0, 0.06)</code>
colorFill	The darkest fill color is used to distinguish between the second and third level of fill color, and is currently only used in the hover effect of Slider.	string	<code>rgba(0, 0, 0, 0.15)</code>
colorIcon	Weak action. Such as `allowClear` or Alert close button	string	<code>rgba(0, 0, 0, 0.45)</code>
colorIconHover	Weak action hover color. Such as `allowClear` or Alert close button	string	<code>rgba(0, 0, 0, 0.88)</code>
colorPrimary	Brand color is one of the most direct visual elements to reflect the characteristics and communication of the product. After you have selected the brand color, we will automatically generate a complete color palette and assign it effective design semantics.	string	<code>#1677ff</code>
colorText	Default text color which comply with W3C standards, and this color is also the darkest neutral color.	string	<code>rgba(0, 0, 0, 0.88)</code>
colorTextLightSolid	Control the highlight color of text with background color, such as the text in Primary Button components.	string	<code>#fff</code>
colorWhite	Pure white color don't changed by theme	string	<code>#fff</code>
borderRadius	Border radius of base components	number	6
borderRadiusLG	LG size border radius, used in some large border radius components, such as Card, Modal and other components.	number	8
borderRadiusOuter	Outer border radius	number	4
borderRadiusSM	SM size border radius, used in small size components, such as Button, Input, Select and other input components in small size	number	4
borderRadiusXS	XS size border radius, used in some small border radius components, such as Segmented, Arrow and other components with small border radius.	number	2
boxShadowTertiary	Control the tertiary box shadow style of an element.	string	0 1px 2px 0 rgba(0, 0, 0, 0.03), 0 1px 6px -1px rgba(0, 0, 0, 0.02), 0 2px 4px 0 rgba(0, 0, 0, 0.02)
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	string	-apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, 'Helvetica Neue', Arial, 'Noto Sans', sans-serif, 'Apple Color Emoji', 'Segoe UI Emoji',

Token Name	Description	Type	Default Value
	browsers, reflecting the friendly, stable and professional characteristics.		'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
fontWeightStrong	Control the font weight of heading components (such as h1, h2, h3) or selected item.	number	600
lineHeight	Line height of text.	number	1.5714285714285714
marginXS	Control the margin of an element, with a small size.	number	8
motionDurationMid	Motion speed, medium speed. Used for medium element animation interaction.	string	0.2s
motionDurationSlow	Motion speed, slow speed. Used for large element animation interaction.	string	0.3s
padding	Control the padding of the element.	number	16
paddingXS	Control the extra small padding of the element.	number	8
sizePopupArrow	The size of the component arrow	number	16
wireframe	Used to change the visual effect of the component to wireframe, if you need to use the V4 effect, you need to enable the configuration item	boolean	false
zIndexPopupBase	Base zIndex of component like FloatButton, Affix which can be cover by large popup	number	1000