When To Use

When part of the page is waiting for asynchronous data or during a rendering process, an appropriate loading animation can effectively alleviate users' inquietude.

Examples

Basic Usage

```
import React from 'react';
import { Spin } from 'antd';

const App: React.FC = () => <Spin />;

export default App;
```

Size

```
import React from 'react';
import { Flex, Spin } from 'antd';

const App: React.FC = () => (
  <Flex align="center" gap="middle">
    <Spin size="small" />
    <Spin />
    <Spin size="large" />
  </Flex>
);

export default App;
```

Embedded mode

```
import React from 'react';
import { Alert, Flex, Spin, Switch } from 'antd';

const App: React.FC = () => {
  const [loading, setLoading] = React.useState<boolean>(false);
  return (
    <Flex gap="middle" vertical>
      <Spin spinning={loading}>
        <Alert
          type="info"
          message="Alert message title"
          description="Further details about the context of this alert."
        />
      </Spin>
      <p>
        Loading state:
        <Switch checked={loading} onChange={setLoading} />
      </p>
    </Flex>
  );
};

export default App;
```

Customized description

```
import React from 'react';
import { Alert, Flex, Spin } from 'antd';

const contentStyle: React.CSSProperties = {
  padding: 50,
  background: 'rgba(0, 0, 0, 0.05)',
  borderRadius: 4,
};

const content = <div style={contentStyle} />;

const App: React.FC = () => (
  <Flex gap="middle" vertical>
    <Flex gap="middle">
      <Spin tip="Loading" size="small">
        {content}
      </Spin>
      <Spin tip="Loading">{content}</Spin>
      <Spin tip="Loading" size="large">
        {content}
      </Spin>
    </Flex>
    <Spin tip="Loading...">
      <Alert
        message="Alert message title"
        description="Further details about the context of this alert."
        type="info"
      />
    </Spin>
  </Flex>
);
```

```
export default App;
```

Delay

```
import React from 'react';
import { Alert, Flex, Spin, Switch } from 'antd';

const App: React.FC = () => {
  const [loading, setLoading] = React.useState<boolean>(false);
  return (
    <Flex gap="middle" vertical>
      <Spin spinning={loading} delay={500}>
        <Alert
          type="info"
          message="Alert message title"
          description="Further details about the context of this alert."
        />
      </Spin>
      <p>
        Loading state:
        <Switch checked={loading} onChange={setLoading} />
      </p>
    </Flex>
  );
};

export default App;
```

Custom spinning indicator

```
import React from 'react';
import { LoadingOutlined } from '@ant-design/icons';
import { Flex, Spin } from 'antd';

const App: React.FC = () => (
  <Flex align="center" gap="middle">
    <Spin indicator={<LoadingOutlined spin />} size="small" />
    <Spin indicator={<LoadingOutlined spin />} />
    <Spin indicator={<LoadingOutlined spin />} size="large" />
    <Spin indicator={<LoadingOutlined style={{ fontSize: 48 }} spin />} />
  </Flex>
);
```

```
export default App;
```

Progress

v5.18.0

```
import React from 'react';
import { Flex, Spin, Switch } from 'antd';

const App: React.FC = () => {
  const [auto, setAuto] = React.useState(false);
  const [percent, setPercent] = React.useState(-50);
  const timerRef = React.useRef<ReturnType<typeof setTimeout>>(null);

  React.useEffect(() => {
    timerRef.current = setTimeout(() => {
      setPercent((v) => {
        const nextPercent = v + 5;
        return nextPercent > 150 ? -50 : nextPercent;
      });
    }, 100);
    return () => clearTimeout(timerRef.current!);
  }, [percent]);

  const mergedPercent = auto ? 'auto' : percent;

  return (
    <Flex align="center" gap="middle">
      <Switch
        checkedChildren="Auto"
        unCheckedChildren="Auto"
        checked={auto}
        onChange={() => {
          setAuto(!auto);
          setPercent(-50);
        }}
      />
      <Spin percent={mergedPercent} size="small" />
      <Spin percent={mergedPercent} />
      <Spin percent={mergedPercent} size="large" />
    </Flex>
  );
};

export default App;
```

Fullscreen

```
import React from 'react';
import { Button, Spin } from 'antd';

const App: React.FC = () => {
  const [spinning, setSpinning] = React.useState(false);
  const [percent, setPercent] = React.useState(0);

  const showLoader = () => {
    setSpinning(true);
    let ptg = -10;

    const interval = setInterval(() => {
      ptg += 5;
      setPercent(ptg);

      if (ptg > 120) {
        clearInterval(interval);
        setSpinning(false);
        setPercent(0);
      }
    }, 100);
  };

  return (
    <>
      <Button onClick={showLoader}>Show fullscreen</Button>
      <Spin spinning={spinning} percent={percent} fullscreen />
    </>
  );
};

export default App;
```

API

Common props ref: Common props

Property	Description	Type	Default	Version
delay	Specifies a delay in milliseconds for loading state (prevent flush)	number (milliseconds)	-	

fullscreen	Display a backdrop with the `Spin` component	boolean	false	5.11.0
indicator	React node of the spinning indicator	ReactNode	-	
percent	The progress percentage, when set to `auto`, it will be an indeterminate progress	number | 'auto'	-	5.18.0
size	The size of Spin, options: `small`, `default` and `large`	string	`default`	
spinning	Whether Spin is visible	boolean	true	
tip	Customize description content when Spin has children	ReactNode	-	
wrapperClassName	The className of wrapper when Spin has children	string	-	

Static Method

- `Spin.setDefaultIndicator(indicator: ReactNode)`

  You can define default spin element globally.

Design Token