何时使用 {#when-to-use}

页面局部处于等待异步数据或正在渲染过程时，合适的加载动效会有效缓解用户的焦虑。

代码演示

基本用法

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

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

export default App;
```

各种大小

```
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;
```

卡片加载中

```
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;
```

自定义描述文案

```
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;
```

延迟

```
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;
```

自定义指示符

```
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;
```

进度

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;
```

全屏

```
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

通用属性参考：通用属性

参数	说明	类型	默认值	版本
delay	延迟显示加载效果的时间（防止闪烁）	number (毫秒)	-	
fullscreen	显示带有 Spin 组件的背景	boolean	false	5.11.0

indicator	加载指示符	ReactNode	-	
percent	展示进度，当设置 `percent="auto"` 时会预估一个永远不会停止的进度	number | 'auto'	-	5.18.0
size	组件大小，可选值为 `small` `default` `large`	string	`default`	
spinning	是否为加载中状态	boolean	true	
tip	当作为包裹元素时，可以自定义描述文案	ReactNode	-	
wrapperClassName	包装器的类属性	string	-	

静态方法

- `Spin.setDefaultIndicator(indicator: ReactNode)`

  你可以自定义全局默认 Spin 的元素。

主题变量（Design Token）