

When To Use

- When there is no data provided, display for friendly tips.
- User tutorial to create something in fresh new situation.

Examples

Basic

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

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

export default App;
```

Choose image

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

const App: React.FC = () => <Empty image={Empty.PRESENTED_IMAGE_SIMPLE} />;

export default App;
```

Customize

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

const App: React.FC = () => (
  <Empty
    image="https://gw.alipayobjects.com/zos/antfincdn/ZHrcdLPrvN/empty.svg"
    styles={{ image: { height: 60 } }}
    description={
      <Typography.Text>
        Customize <a href="#API">Description</a>
      </Typography.Text>
    }
  >
    <Button type="primary">Create Now</Button>
  </Empty>
);

export default App;
```

ConfigProvider

```
import React, { useState } from 'react';
import { SmileOutlined } from '@ant-design/icons';
import {
  Cascader,
  ConfigProvider,
  Divider,
  List,
  Select,
  Space,
  Switch,
  Table,
  Transfer,
  TreeSelect,
} from 'antd';

const customizeRenderEmpty = () => (
  <div style={{ textAlign: 'center' }}>
    <SmileOutlined style={{ fontSize: 20 }} />
    <p>Data Not Found</p>
  </div>
);

const style: React.CSSProperties = { width: 200 };

const App: React.FC = () => {
  const [customize, setCustomize] = useState(true);
  return (
    <>
      <Switch
        uncheckedChildren="default"
        checkedChildren="customize"
        checked={customize}
        onChange={setCustomize}
      />
      <Divider />
      <ConfigProvider renderEmpty={customize ? customizeRenderEmpty :
undefined}>
        <Space direction="vertical" style={{ width: '100%' }}>
          <h4>Select</h4>
          <Select style={style} />
          <h4>TreeSelect</h4>
          <TreeSelect style={style} treeData={[]} />
          <h4>Cascader</h4>
          <Cascader style={style} options={[]} showSearch />
        </Space>
      </>
    </>
  );
};
```

```

    <h4>Transfer</h4>
    <Transfer />
    <h4>Table</h4>
    <Table
      style={{ marginTop: 8 }}
      columns={[
        { title: 'Name', dataIndex: 'name', key: 'name' },
        { title: 'Age', dataIndex: 'age', key: 'age' },
      ]}
    />
    <h4>List</h4>
    <List />
  </Space>
</ConfigProvider>
</>
);
};

export default App;

```

No description

```

import React from 'react';
import { Empty } from 'antd';

const App: React.FC = () => <Empty description={false} />;

export default App;

```

API

Common props ref: [Common props](#)

```

<Empty>
  <Button>Create</Button>
</Empty>

```

Property	Description	Type	Default	Version
description	Customize description	ReactNode	-	
image	Customize image. Will treat as	ReactNode	Empty.PRESENTED_IMAGE_DEFAULT	

	image url when string provided			
imageStyle	The style of image	CSSProperties	-	

Built-in images

- Empty.PRESENTED_IMAGE_SIMPLE



- Empty.PRESENTED_IMAGE_DEFAULT



Semantic DOM

演示

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

import SemanticPreview from '../../../../dumi/components/SemanticPreview';
import useLocale from '../../../../dumi/hooks/useLocale';

const locales = {
  cn: {
    root: '根元素',
    image: '图标元素',
    description: '描述元素',
    footer: '底部元素',
  },
  en: {
    root: 'Root element',
    image: 'Image element',
    description: 'Description element',
    footer: 'Footer element',
  },
};
```

```

const App: React.FC = () => {
  const [locale] = useLocale(locales);
  return (
    <SemanticPreview
      componentName="Empty"
      semantics={[
        { name: 'root', desc: locale.root, version: '5.23.0' },
        { name: 'image', desc: locale.image, version: '5.23.0' },
        { name: 'description', desc: locale.description, version: '5.23.0'
      },
      { name: 'footer', desc: locale.footer, version: '5.23.0' },
    ]}
    >
    <Empty
      image={Empty.PRESENTED_IMAGE_SIMPLE}
      styles={{ image: { height: 60 } }}
      description={
        <Typography.Text>
          Customize <a href="#API">Description</a>
        </Typography.Text>
      }
    >
    <Button type="primary">Create Now</Button>
  </Empty>
</SemanticPreview>
);
};

export default App;

```

Design Token