何时使用 {#when-to-use}

最基础的列表展示，可承载文字、列表、图片、段落，常用于后台数据展示页面。

代码演示

简单列表

```
import React from 'react';
import { Divider, List, Typography } from 'antd';

const data = [
  'Racing car sprays burning fuel into crowd.',
  'Japanese princess to wed commoner.',
  'Australian walks 100km after outback crash.',
  'Man charged over missing wedding girl.',
  'Los Angeles battles huge wildfires.',
];

const App: React.FC = () => (
  <>
    <Divider orientation="left">Default Size</Divider>
    <List
      header={<div>Header</div>}
      footer={<div>Footer</div>}
      bordered
      dataSource={data}
      renderItem={(item) => (
        <List.Item>
          <Typography.Text mark>[ITEM]</Typography.Text> {item}
        </List.Item>
      )}
    />
    <Divider orientation="left">Small Size</Divider>
    <List
      size="small"
      header={<div>Header</div>}
      footer={<div>Footer</div>}
      bordered
      dataSource={data}
      renderItem={(item) => <List.Item>{item}</List.Item>}
    />
    <Divider orientation="left">Large Size</Divider>
    <List
      size="large"
      header={<div>Header</div>}
```

```
      footer={<div>Footer</div>}
      bordered
      dataSource={data}
      renderItem={(item) => <List.Item>{item}</List.Item>}
    />
  </>
);

export default App;
```

基础列表

```
import React from 'react';
import { Avatar, List } from 'antd';

const data = [
  {
    title: 'Ant Design Title 1',
  },
  {
    title: 'Ant Design Title 2',
  },
  {
    title: 'Ant Design Title 3',
  },
  {
    title: 'Ant Design Title 4',
  },
];

const App: React.FC = () => (
  <List
    itemLayout="horizontal"
    dataSource={data}
    renderItem={(item, index) => (
      <List.Item>
        <List.Item.Meta
          avatar={<Avatar src={`https://api.dicebear.com/7.x/miniavs/svg?
seed=${index}`} />}
          title={<a href="https://ant.design">{item.title}</a>}
          description="Ant Design, a design language for background
applications, is refined by Ant UED Team"
        />
      </List.Item>
    )}
  />
```

```
);

export default App;
```

加载更多

```
import React, { useEffect, useState } from 'react';
import { Avatar, Button, List, Skeleton } from 'antd';

interface DataType {
  gender?: string;
  name: {
    title?: string;
    first?: string;
    last?: string;
  };
  email?: string;
  picture: {
    large?: string;
    medium?: string;
    thumbnail?: string;
  };
  nat?: string;
  loading: boolean;
}

const count = 3;
const fakeDataUrl = `https://randomuser.me/api/?
results=${count}&inc=name,gender,email,nat,picture&noinfo`;

const App: React.FC = () => {
  const [initLoading, setInitLoading] = useState(true);
  const [loading, setLoading] = useState(false);
  const [data, setData] = useState<DataType[]>([]);
  const [list, setList] = useState<DataType[]>([]);

  useEffect(() => {
    fetch(fakeDataUrl)
      .then((res) => res.json())
      .then((res) => {
        setInitLoading(false);
        setData(res.results);
        setList(res.results);
      });
  }, []);
```

```
  const onLoadMore = () => {
    setLoading(true);
    setList(
      data.concat(
        Array.from({ length: count }).map(() => ({ loading: true, name: {},
picture: {} })),
      ),
    );
    fetch(fakeDataUrl)
      .then((res) => res.json())
      .then((res) => {
        const newData = data.concat(res.results);
        setData(newData);
        setList(newData);
        setLoading(false);
        // Resetting window's offsetTop so as to display react-virtualized
demo underfloor.
        // In real scene, you can using public method of react-virtualized:
        // https://stackoverflow.com/questions/46700726/how-to-use-public-
method-updateposition-of-react-virtualized
        window.dispatchEvent(new Event('resize'));
      });
  };

  const loadMore =
    !initLoading && !loading ? (
      <div
        style={{
          textAlign: 'center',
          marginTop: 12,
          height: 32,
          lineHeight: '32px',
        }}
      >
        <Button onClick={onLoadMore}>loading more</Button>
      </div>
    ) : null;

  return (
    <List
      className="demo-loadmore-list"
      loading={initLoading}
      itemLayout="horizontal"
      loadMore={loadMore}
      dataSource={list}
      renderItem={(item) => (
```

```
        <List.Item
          actions={[<a key="list-loadmore-edit">edit</a>, <a key="list-
loadmore-more">more</a>]}
        >
          <Skeleton avatar title={false} loading={item.loading} active>
            <List.Item.Meta
              avatar={<Avatar src={item.picture.large} />}
              title={<a href="https://ant.design">{item.name?.last}</a>}
              description="Ant Design, a design language for background
applications, is refined by Ant UED Team"
            />
            <div>content</div>
          </Skeleton>
        </List.Item>
      )}
    />
  );
};

export default App;
```

竖排列表样式

```
import React from 'react';
import { LikeOutlined, MessageOutlined, StarOutlined } from '@ant-
design/icons';
import { Avatar, List, Space } from 'antd';

const data = Array.from({ length: 23 }).map((_, i) => ({
  href: 'https://ant.design',
  title: `ant design part ${i}`,
  avatar: `https://api.dicebear.com/7.x/miniavs/svg?seed=${i}`,
  description:
    'Ant Design, a design language for background applications, is refined
by Ant UED Team.',
  content:
    'We supply a series of design principles, practical patterns and high
quality design resources (Sketch and Axure), to help people create their
product prototypes beautifully and efficiently.',
}));

const IconText = ({ icon, text }: { icon: React.FC; text: string }) => (
  <Space>
    {React.createElement(icon)}
    {text}
  </Space>
```

```
);

const App: React.FC = () => (
  <List
    itemLayout="vertical"
    size="large"
    pagination={{
      onChange: (page) => {
        console.log(page);
      },
      pageSize: 3,
    }}
    dataSource={data}
    footer={
      <div>
        <b>ant design</b> footer part
      </div>
    }
    renderItem={(item) => (
      <List.Item
        key={item.title}
        actions={[
          <IconText icon={StarOutlined} text="156" key="list-vertical-star-
o" />,
          <IconText icon={LikeOutlined} text="156" key="list-vertical-like-
o" />,
          <IconText icon={MessageOutlined} text="2" key="list-vertical-
message" />,
        ]}
        extra={
          <img
            width={272}
            alt="logo"

src="https://gw.alipayobjects.com/zos/rmsportal/mqaQswcyDLcXyDKnZfES.png"
          />
        }
      >
        <List.Item.Meta
          avatar={<Avatar src={item.avatar} />}
          title={<a href={item.href}>{item.title}</a>}
          description={item.description}
        />
        {item.content}
      </List.Item>
    )}
```

```
  />
);

export default App;
```

分页设置

```
import React, { useState } from 'react';
import { Avatar, List, Radio, Space } from 'antd';

type PaginationPosition = 'top' | 'bottom' | 'both';

type PaginationAlign = 'start' | 'center' | 'end';

const data = [
  {
    title: 'Ant Design Title 1',
  },
  {
    title: 'Ant Design Title 2',
  },
  {
    title: 'Ant Design Title 3',
  },
  {
    title: 'Ant Design Title 4',
  },
];

const positionOptions = ['top', 'bottom', 'both'];

const alignOptions = ['start', 'center', 'end'];

const App: React.FC = () => {
  const [position, setPosition] = useState<PaginationPosition>('bottom');
  const [align, setAlign] = useState<PaginationAlign>('center');

  return (
    <>
      <Space direction="vertical" style={{ marginBottom: '20px' }}
size="middle">
        <Space>
          <span>Pagination Position:</span>
          <Radio.Group
            optionType="button"
            value={position}
```

```
            onChange={(e) => {
              setPosition(e.target.value);
            }}
          >
            {positionOptions.map((item) => (
              <Radio.Button key={item} value={item}>
                {item}
              </Radio.Button>
            ))}
          </Radio.Group>
        </Space>
        <Space>
          <span>Pagination Align:</span>
          <Radio.Group
            optionType="button"
            value={align}
            onChange={(e) => {
              setAlign(e.target.value);
            }}
          >
            {alignOptions.map((item) => (
              <Radio.Button key={item} value={item}>
                {item}
              </Radio.Button>
            ))}
          </Radio.Group>
        </Space>
      </Space>
      <List
        pagination={{ position, align }}
        dataSource={data}
        renderItem={(item, index) => (
          <List.Item>
            <List.Item.Meta
              avatar={<Avatar src=
{`https://api.dicebear.com/7.x/miniavs/svg?seed=${index}`} />}
              title={<a href="https://ant.design">{item.title}</a>}
              description="Ant Design, a design language for background
applications, is refined by Ant UED Team"
            />
          </List.Item>
        )}
      />
    </>
  );
};
```

```
export default App;
```

栅格列表

```
import React from 'react';
import { Card, List } from 'antd';

const data = [
  {
    title: 'Title 1',
  },
  {
    title: 'Title 2',
  },
  {
    title: 'Title 3',
  },
  {
    title: 'Title 4',
  },
];

const App: React.FC = () => (
  <List
    grid={{ gutter: 16, column: 4 }}
    dataSource={data}
    renderItem={(item) => (
      <List.Item>
        <Card title={item.title}>Card content</Card>
      </List.Item>
    )}
  />
);

export default App;
```

测试栅格列表

Debug

```
import React from 'react';
import { Card, List } from 'antd';

const data = [
  {
```

```
    title: 'Title 1',
  },
  {
    title: 'Title 2',
  },
  {
    title: 'Title 3',
  },
  {
    title: 'Title 4',
  },
  {
    title: 'Title 5',
  },
  {
    title: 'Title 6',
  },
];

const ListItem = () => (
  <List.Item>
    <Card title="title">Card content</Card>
  </List.Item>
);

const App: React.FC = () => (
  <>
    <List
      grid={{ gutter: 16, column: 4 }}
      dataSource={data}
      renderItem={(item) => (
        <List.Item>
          <Card title={item.title}>Card content</Card>
        </List.Item>
      )}
    />
    <List grid={{ gutter: 16, column: 4 }} dataSource={data} renderItem={()
=> <ListItem />} />
    <List
      grid={{ gutter: 16, column: 4 }}
      dataSource={data}
      renderItem={() => (
        <>
          <ListItem />
          <div />
        </>
```

```
      )}
    />
  </>
);

export default App;
```

响应式的栅格列表

```
import React from 'react';
import { Card, List } from 'antd';

const data = [
  {
    title: 'Title 1',
  },
  {
    title: 'Title 2',
  },
  {
    title: 'Title 3',
  },
  {
    title: 'Title 4',
  },
  {
    title: 'Title 5',
  },
  {
    title: 'Title 6',
  },
];

const App: React.FC = () => (
  <List
    grid={{
      gutter: 16,
      xs: 1,
      sm: 2,
      md: 4,
      lg: 4,
      xl: 6,
      xxl: 3,
    }}
    dataSource={data}
    renderItem={(item) => (
```

```
      <List.Item>
        <Card title={item.title}>Card content</Card>
      </List.Item>
    )}
  />
);

export default App;
```

滚动加载

```
import React, { useEffect, useState } from 'react';
import { Avatar, Divider, List, Skeleton } from 'antd';
import InfiniteScroll from 'react-infinite-scroll-component';

interface DataType {
  gender: string;
  name: {
    title: string;
    first: string;
    last: string;
  };
  email: string;
  picture: {
    large: string;
    medium: string;
    thumbnail: string;
  };
  nat: string;
}

const App: React.FC = () => {
  const [loading, setLoading] = useState(false);
  const [data, setData] = useState<DataType[]>([]);

  const loadMoreData = () => {
    if (loading) {
      return;
    }
    setLoading(true);
    fetch('https://randomuser.me/api/?
results=10&inc=name,gender,email,nat,picture&noinfo')
      .then((res) => res.json())
      .then((body) => {
        setData([...data, ...body.results]);
        setLoading(false);
```

```
      })
      .catch(() => {
        setLoading(false);
      });
  };

  useEffect(() => {
    loadMoreData();
  }, []);

  return (
    <div
      id="scrollableDiv"
      style={{
        height: 400,
        overflow: 'auto',
        padding: '0 16px',
        border: '1px solid rgba(140, 140, 140, 0.35)',
      }}
    >
      <InfiniteScroll
        dataLength={data.length}
        next={loadMoreData}
        hasMore={data.length < 50}
        loader={<Skeleton avatar paragraph={{ rows: 1 }} active />}
        endMessage={<Divider plain>It is all, nothing more 🤐</Divider>}
        scrollableTarget="scrollableDiv"
      >
        <List
          dataSource={data}
          renderItem={(item) => (
            <List.Item key={item.email}>
              <List.Item.Meta
                avatar={<Avatar src={item.picture.large} />}
                title={<a href="https://ant.design">{item.name.last}</a>}
                description={item.email}
              />
              <div>Content</div>
            </List.Item>
          )}
        />
      </InfiniteScroll>
    </div>
  );
};
```

```
export default App;
```

滚动加载无限长列表

```
import React, { useEffect, useState } from 'react';
import { Avatar, List, message } from 'antd';
import VirtualList from 'rc-virtual-list';

interface UserItem {
  email: string;
  gender: string;
  name: {
    first: string;
    last: string;
    title: string;
  };
  nat: string;
  picture: {
    large: string;
    medium: string;
    thumbnail: string;
  };
}

const fakeDataUrl =
  'https://randomuser.me/api/?
results=20&inc=name,gender,email,nat,picture&noinfo';
const ContainerHeight = 400;

const App: React.FC = () => {
  const [data, setData] = useState<UserItem[]>([]);

  const appendData = (showMessage = true) => {
    fetch(fakeDataUrl)
      .then((res) => res.json())
      .then((body) => {
        setData(data.concat(body.results));
        showMessage && message.success(`${body.results.length} more items
loaded!`);
      });
  };

  useEffect(() => {
    appendData(false);
  }, []);
```

```
  const onScroll = (e: React.UIEvent<HTMLElement, UIEvent>) => {
    // Refer to: https://developer.mozilla.org/en-
US/docs/Web/API/Element/scrollHeight#problems_and_solutions
    if (Math.abs(e.currentTarget.scrollHeight - e.currentTarget.scrollTop -
ContainerHeight) <= 1) {
      appendData();
    }
  };

  return (
    <List>
      <VirtualList
        data={data}
        height={ContainerHeight}
        itemHeight={47}
        itemKey="email"
        onScroll={onScroll}
      >
        {(item: UserItem) => (
          <List.Item key={item.email}>
            <List.Item.Meta
              avatar={<Avatar src={item.picture.large} />}
              title={<a href="https://ant.design">{item.name.last}</a>}
              description={item.email}
            />
            <div>Content</div>
          </List.Item>
        )}
      </VirtualList>
    </List>
  );
};

export default App;
```

自定义组件 token

Debug

```
import React from 'react';
import { Avatar, ConfigProvider, Divider, List, Typography } from 'antd';

const data = [
  'Racing car sprays burning fuel into crowd.',
  'Japanese princess to wed commoner.',
```

```
  'Australian walks 100km after outback crash.',
  'Man charged over missing wedding girl.',
  'Los Angeles battles huge wildfires.',
];

const data1 = [
  {
    title: 'Ant Design Title 1',
  },
  {
    title: 'Ant Design Title 2',
  },
  {
    title: 'Ant Design Title 3',
  },
  {
    title: 'Ant Design Title 4',
  },
];

const App: React.FC = () => (
  <ConfigProvider
    theme={{
      components: {
        List: {
          headerBg: 'pink',
          footerBg: 'pink',
          emptyTextPadding: 32,
          itemPadding: '26px',
          itemPaddingSM: '16px',
          itemPaddingLG: '36px',
          metaMarginBottom: 20,
          avatarMarginRight: 20,
          titleMarginBottom: 10,
          descriptionFontSize: 20,
        },
      },
    }}
  >
    <Divider orientation="left">Default Size</Divider>
    <List
      header={<div>Header</div>}
      footer={<div>Footer</div>}
      bordered
      dataSource={data}
      renderItem={(item) => (
```

```
        <List.Item>
          <Typography.Text mark>[ITEM]</Typography.Text> {item}
        </List.Item>
      )}
    />
    <Divider orientation="left">Small Size</Divider>
    <List
      size="small"
      header={<div>Header</div>}
      footer={<div>Footer</div>}
      bordered
      dataSource={data}
      renderItem={(item) => <List.Item>{item}</List.Item>}
    />
    <Divider orientation="left">Large Size</Divider>
    <List
      size="large"
      header={<div>Header</div>}
      footer={<div>Footer</div>}
      bordered
      dataSource={data}
      renderItem={(item) => <List.Item>{item}</List.Item>}
    />
    <Divider orientation="left">Meta</Divider>
    <List
      itemLayout="horizontal"
      dataSource={data1}
      renderItem={(item, index) => (
        <List.Item>
          <List.Item.Meta
            avatar={<Avatar src={`https://api.dicebear.com/7.x/miniavs/svg?
seed=${index}`} />}
            title={<a href="https://ant.design">{item.title}</a>}
            description="Ant Design, a design language for background
applications, is refined by Ant UED Team"
          />
        </List.Item>
      )}
    />
    <Divider orientation="left">Vertical</Divider>
    <List
      itemLayout="vertical"
      dataSource={data1}
      renderItem={(item, index) => (
        <List.Item>
          <List.Item.Meta
```

```
            avatar={<Avatar src={`https://api.dicebear.com/7.x/miniavs/svg?
seed=${index}`} />}
            title={<a href="https://ant.design">{item.title}</a>}
            description="Ant Design, a design language for background
applications, is refined by Ant UED Team"
          />
        </List.Item>
      )}
    />
    <Divider orientation="left">Empty Text</Divider>
    <List />
  </ConfigProvider>
);

export default App;
```

API

通用属性参考：通用属性

另外我们封装了 ProList，在 `antd` List 之上扩展了更多便捷易用的功能，比如多选，展开等功能，使用体验贴近 Table，欢迎尝试使用。

List

参数	说明	类型	默认值	版本
bordered	是否展示边框	boolean	false	
dataSource	列表数据源	any[]	-	
footer	列表底部	ReactNode	-	
grid	列表栅格配置	object	-	
header	列表头部	ReactNode	-	
itemLayout	设置 `List.Item` 布局，设置成 `vertical` 则竖直样式显示，默认横排	string	-	
loading	当卡片内容还在加载中时，可以用 `loading` 展示一个占位	boolean | object (更多)	false	
loadMore	加载更多	ReactNode	-	
locale	默认文案设置，目前包括空数据文案	object	{emptyText: 暂无数据}	

pagination	对应的 `pagination` 配置，设置 false 不显示	boolean | object	false	
renderItem	当使用 dataSource 时，可以用 `renderItem` 自定义渲染列表项	(item: T, index: number) => ReactNode	-	
rowKey	当 `renderItem` 自定义渲染列表项有效时，自定义每一行的 key 的获取方式	keyof T | (item: T) => `React.Key`	`"key"`	
size	list 的尺寸	`default` | `large` | `small`	`default`	
split	是否展示分割线	boolean	true	

pagination

分页的配置项。

参数	说明	类型	默认值
position	指定分页显示的位置	`top` | `bottom` | `both`	`bottom`
align	指定分页对齐的位置	`start` | `center` | `end`	`end`

更多配置项，请查看 `Pagination` 。

List grid props

参数	说明	类型	默认值	版本
column	列数	number	-	
gutter	栅格间隔	number	0	
xs	<576px 展示的列数	number	-	
sm	≥576px 展示的列数	number	-	
md	≥768px 展示的列数	number	-	
lg	≥992px 展示的列数	number	-	
xl	≥1200px 展示的列数	number	-	
xxl	≥1600px 展示的列数	number	-	

List.Item

参数	说明	类型	默认	版本

			值	
actions	列表操作组，根据 itemLayout 的不同，位置在卡片底部或者最右侧	Array<ReactNode>	-	
classNames	语义化结构 className	Record<actions | extra, string>	-	5.18.0
extra	额外内容，通常用在 itemLayout 为 vertical 的情况下，展示右侧内容; horizontal 展示在列表元素最右侧	ReactNode	-	
styles	语义化结构 style	Record<actions | extra, CSSProperties>	-	5.18.0

List.Item.Meta

参数	说明	类型	默认值	版本
avatar	列表元素的图标	ReactNode	-	
description	列表元素的描述内容	ReactNode	-	
title	列表元素的标题	ReactNode	-	

Semantic DOM

演示

```
import React from 'react';

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

import { Avatar, List, Space } from 'antd';
import { LikeOutlined, MessageOutlined, StarOutlined } from '@ant-
design/icons';

const locales = {
  cn: {
    extra: '设置额外内容',
    actions: '设置列表操作组',
  },
  en: {
    extra: 'set `extra` of List.Item',
    actions: 'set `actions` of List.Item',
```

```
  },
};

const IconText = ({ icon, text }: { icon: React.FC; text: string }) => (
  <Space>
    {React.createElement(icon)}
    {text}
  </Space>
);

const data = Array.from({ length: 1 }).map((_, i) => ({
  href: 'https://ant.design',
  title: `ant design part ${i}`,
  avatar: `https://api.dicebear.com/7.x/miniavs/svg?seed=${i}`,
  description:
    'Ant Design, a design language for background applications, is refined
by Ant UED Team.',
  content:
    'We supply a series of design principles, practical patterns and high
quality design resources (Sketch and Axure), to help people create their
product prototypes beautifully and efficiently.',
}));

const BlockList: React.FC<React.PropsWithChildren> = (props) => {
  const divRef = React.useRef<HTMLDivElement>(null);

  return (
    <div ref={divRef} style={{ position: 'absolute', inset: 0, height: 300
}}>
      <List
        itemLayout="vertical"
        size="large"
        dataSource={data}
        renderItem={(item) => (
          <List.Item
            {...props}
            key={item.title}
            actions={[
              <IconText icon={StarOutlined} text="156" key="list-vertical-
star-o" />,
              <IconText icon={LikeOutlined} text="156" key="list-vertical-
like-o" />,
              <IconText icon={MessageOutlined} text="2" key="list-vertical-
message" />,
            ]}
            extra={
```

```
              <img
                width={272}
                alt="logo"

src="https://gw.alipayobjects.com/zos/rmsportal/mqaQswcyDLcXyDKnZfES.png"
              />
            }
          >
            <List.Item.Meta
              avatar={<Avatar src={item.avatar} />}
              title={<a href={item.href}>{item.title}</a>}
              description={item.description}
            />
            {item.content}
          </List.Item>
        )}
      />
    </div>
  );
};

const App: React.FC = () => {
  const [locale] = useLocale(locales);
  return (
    <SemanticPreview
      componentName="List"
      height={300}
      semantics={[
        { name: 'extra', desc: locale.extra, version: '5.18.0' },
        { name: 'actions', desc: locale.actions, version: '5.18.0' },
      ]}
    >
      <BlockList />
    </SemanticPreview>
  );
};

export default App;
```

主题变量（Design Token）