

When To Use

When a given task is complicated or has a certain sequence in the series of subtasks, we can decompose it into several steps to make things easier.

Examples

Basic

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

const description = 'This is a description.';
const App: React.FC = () => (
  <Steps
    current={1}
    items={[
      {
        title: 'Finished',
        description,
      },
      {
        title: 'In Progress',
        description,
        subTitle: 'Left 00:00:08',
      },
      {
        title: 'Waiting',
        description,
      },
    ]}
  />
);

export default App;
```

Mini version

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

const App: React.FC = () => (
  <Steps
    size="small"
    current={1}
    items={[]
```

```

    {
      title: 'Finished',
    },
    {
      title: 'In Progress',
    },
    {
      title: 'Waiting',
    },
  ],
]
  />
);

export default App;

```

With icon

```

import React from 'react';
import { LoadingOutlined, SmileOutlined, SolutionOutlined, UserOutlined }
from '@ant-design/icons';
import { Steps } from 'antd';

const App: React.FC = () => (
  <Steps
    items={[
      {
        title: 'Login',
        status: 'finish',
        icon: <UserOutlined />,
      },
      {
        title: 'Verification',
        status: 'finish',
        icon: <SolutionOutlined />,
      },
      {
        title: 'Pay',
        status: 'process',
        icon: <LoadingOutlined />,
      },
      {
        title: 'Done',
        status: 'wait',
        icon: <SmileOutlined />,
      },
    ]}
  />
);

```

```
    />
  );

  export default App;
```

Switch Step

```
import React, { useState } from 'react';
import { Button, message, Steps, theme } from 'antd';

const steps = [
  {
    title: 'First',
    content: 'First-content',
  },
  {
    title: 'Second',
    content: 'Second-content',
  },
  {
    title: 'Last',
    content: 'Last-content',
  },
];

const App: React.FC = () => {
  const { token } = theme.useToken();
  const [current, setCurrent] = useState(0);

  const next = () => {
    setCurrent(current + 1);
  };

  const prev = () => {
    setCurrent(current - 1);
  };

  const items = steps.map((item) => ({ key: item.title, title: item.title }));

  const contentStyle: React.CSSProperties = {
    lineHeight: '260px',
    textAlign: 'center',
    color: token.colorTextTertiary,
    backgroundColor: token.colorFillAlter,
    borderRadius: token.borderRadiusLG,
```

```

    border: `1px dashed ${token.colorBorder}`,
    marginTop: 16,
  };

  return (
    <>
      <Steps current={current} items={items} />
      <div style={contentStyle}>{steps[current].content}</div>
      <div style={{ marginTop: 24 }}>
        {current < steps.length - 1 && (
          <Button type="primary" onClick={() => next()}>
            Next
          </Button>
        )}
        {current === steps.length - 1 && (
          <Button type="primary" onClick={() => message.success('Processing
complete!')}>
            Done
          </Button>
        )}
        {current > 0 && (
          <Button style={{ margin: '0 8px' }} onClick={() => prev()}>
            Previous
          </Button>
        )}
      </div>
    </>
  );
};

export default App;

```

Vertical

```

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

const description = 'This is a description.';
const App: React.FC = () => (
  <Steps
    direction="vertical"
    current={1}
    items={[
      {
        title: 'Finished',
        description,

```

```

    },
    {
      title: 'In Progress',
      description,
    },
    {
      title: 'Waiting',
      description,
    },
  ],
]
  />
);

export default App;

```

Vertical mini version

```

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

const description = 'This is a description.';
const App: React.FC = () => (
  <Steps
    direction="vertical"
    size="small"
    current={1}
    items={[
      { title: 'Finished', description },
      {
        title: 'In Progress',
        description,
      },
      {
        title: 'Waiting',
        description,
      },
    ]}
  />
);

export default App;

```

Error status

```

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

const description = 'This is a description';
const App: React.FC = () => (
  <Steps
    current={1}
    status="error"
    items={[
      {
        title: 'Finished',
        description,
      },
      {
        title: 'In Process',
        description,
      },
      {
        title: 'Waiting',
        description,
      },
    ]}
  />
);

export default App;

```

Dot Style

```

import React from 'react';
import { Divider, Steps } from 'antd';

const App: React.FC = () => (
  <>
    <Steps
      progressDot
      current={1}
      items={[
        {
          title: 'Finished',
          description: 'This is a description.',
        },
        {
          title: 'In Progress',
          description: 'This is a description.',
        },
      ]}
    />
  </>
);

```

```

    },
    {
      title: 'Waiting',
      description: 'This is a description.',
    },
  ]}
/>
<Divider />
<Steps
  progressDot
  current={1}
  direction="vertical"
  items={[
    {
      title: 'Finished',
      description: 'This is a description. This is a description.',
    },
    {
      title: 'Finished',
      description: 'This is a description. This is a description.',
    },
    {
      title: 'In Progress',
      description: 'This is a description. This is a description.',
    },
    {
      title: 'Waiting',
      description: 'This is a description.',
    },
    {
      title: 'Waiting',
      description: 'This is a description.',
    },
  ]}
/>
</>
);

export default App;

```

Customized Dot Style

```

import React from 'react';
import type { StepsProps } from 'antd';
import { Popover, Steps } from 'antd';

```

```

const customDot: StepsProps['progressDot'] = (dot, { status, index }) => (
  <Popover
    content={
      <span>
        step {index} status: {status}
      </span>
    }
  >
    {dot}
  </Popover>
);
const description = 'You can hover on the dot.';
const App: React.FC = () => (
  <Steps
    current={1}
    progressDot={customDot}
    items={[
      {
        title: 'Finished',
        description,
      },
      {
        title: 'In Progress',
        description,
      },
      {
        title: 'Waiting',
        description,
      },
      {
        title: 'Waiting',
        description,
      },
    ]}
  />
);

export default App;

```

Dot Style Size Small

Debug

```

import React from 'react';
import { Divider, Steps } from 'antd';

```



```
const App: React.FC = () => (  
  <>  
    <Steps  
      progressDot  
      current={1}  
      size="small"  
      items={[  
        {  
          title: 'Finished',  
          description: 'This is a description.',  
        },  
        {  
          title: 'In Progress',  
          description: 'This is a description.',  
        },  
        {  
          title: 'Waiting',  
          description: 'This is a description.',  
        },  
      ]}  
    />  
    <Divider />  
    <Steps  
      progressDot  
      current={1}  
      direction="vertical"  
      size="small"  
      items={[  
        {  
          title: 'Finished',  
          description: 'This is a description. This is a description.',  
        },  
        {  
          title: 'Finished',  
          description: 'This is a description. This is a description.',  
        },  
        {  
          title: 'In Progress',  
          description: 'This is a description. This is a description.',  
        },  
        {  
          title: 'Waiting',  
          description: 'This is a description.',  
        },  
        {  
          title: 'Waiting',  

```

```

        description: 'This is a description.',
      },
    ]}
  />
</>
);

export default App;

```

Clickable

```

import React, { useState } from 'react';
import { Divider, Steps } from 'antd';

const App: React.FC = () => {
  const [current, setCurrent] = useState(0);

  const onChange = (value: number) => {
    console.log('onChange:', value);
    setCurrent(value);
  };
  const description = 'This is a description.';

  return (
    <>
      <Steps
        current={current}
        onChange={onChange}
        items={[
          {
            title: 'Step 1',
            description,
          },
          {
            title: 'Step 2',
            description,
          },
          {
            title: 'Step 3',
            description,
          },
        ]}
      />

      <Divider />
    </>
  );
}

```

```

    <Steps
      current={current}
      onChange={onChange}
      direction="vertical"
      items={[
        {
          title: 'Step 1',
          description,
        },
        {
          title: 'Step 2',
          description,
        },
        {
          title: 'Step 3',
          description,
        },
      ]}
    />
  </>
);
};

export default App;

```

Navigation Steps

```

import React, { useState } from 'react';
import { Steps } from 'antd';

const App: React.FC = () => {
  const [current, setCurrent] = useState(0);

  const onChange = (value: number) => {
    console.log('onChange:', value);
    setCurrent(value);
  };

  return (
    <>
      <Steps
        type="navigation"
        size="small"
        current={current}
        onChange={onChange}
        className="site-navigation-steps"

```

```

items=[
  {
    title: 'Step 1',
    subTitle: '00:00:05',
    status: 'finish',
    description: 'This is a description.',
  },
  {
    title: 'Step 2',
    subTitle: '00:01:02',
    status: 'process',
    description: 'This is a description.',
  },
  {
    title: 'Step 3',
    subTitle: 'waiting for longlong time',
    status: 'wait',
    description: 'This is a description.',
  },
]
})
/>
<Steps
  type="navigation"
  current={current}
  onChange={onChange}
  className="site-navigation-steps"
  items=[
    {
      status: 'finish',
      title: 'Step 1',
    },
    {
      status: 'process',
      title: 'Step 2',
    },
    {
      status: 'wait',
      title: 'Step 3',
    },
    {
      status: 'wait',
      title: 'Step 4',
    },
  ],
]
/>
<Steps

```

```

    type="navigation"
    size="small"
    current={current}
    onChange={onChange}
    className="site-navigation-steps"
    items={[
      {
        status: 'finish',
        title: 'finish 1',
      },
      {
        status: 'finish',
        title: 'finish 2',
      },
      {
        status: 'process',
        title: 'current process',
      },
      {
        status: 'wait',
        title: 'wait',
        disabled: true,
      },
    ]}
  />
</>
);
};

export default App;

```

Steps with progress

```

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

const description = 'This is a description.';
const App: React.FC = () => (
  <Steps
    current={1}
    percent={60}
    items={[
      {
        title: 'Finished',
        description,
      },
    ]}
  />

```

```

        {
          title: 'In Progress',
          subTitle: 'Left 00:00:08',
          description,
        },
        {
          title: 'Waiting',
          description,
        },
      ],
    ]}
  />
);

export default App;

```

Label Placement

```

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

const description = 'This is a description.';
const items = [
  {
    title: 'Finished',
    description,
  },
  {
    title: 'In Progress',
    description,
  },
  {
    title: 'Waiting',
    description,
  },
];

const App: React.FC = () => (
  <>
    <Steps current={1} labelPlacement="vertical" items={items} />
    <br />
    <Steps current={1} percent={60} labelPlacement="vertical" items={items} />
  />
  <br />
  <Steps current={1} percent={80} size="small" labelPlacement="vertical"
items={items} />
</>
);

```

```
export default App;
```

Progress Debug

Debug

```
import React, { useState } from 'react';
import type { StepsProps } from 'antd';
import { Button, Space, Steps } from 'antd';

const App: React.FC = () => {
  const [percent, setPercentage] = useState<number | undefined>(0);
  const [current, setCurrent] = useState(1);
  const [status, setStatus] = useState<StepsProps['status']>('process');
  const description = 'This is a description.';
  const items = [
    {
      title: 'Finished',
      description,
    },
    {
      title: 'In Progress',
      subTitle: 'Left 00:00:08',
      description,
    },
    {
      title: 'Waiting',
      description,
    },
  ];
  return (
    <>
      <Space.Compact block>
        <Button onClick={() => setPercentage(undefined)}>Percentage to
undefined</Button>
        <Button onClick={() => setPercentage((prev) => ((prev ?? 0) + 10) %
100)}>
          Percentage +
        </Button>
        <Button onClick={() => setCurrent((prev) => (prev + 1) %
3)}>Current +</Button>
        <Button onClick={() => setStatus('wait')}>Status Wait</Button>
        <Button onClick={() => setStatus('process')}>Status
Process</Button>
        <Button onClick={() => setStatus('finish')}>Status Finish</Button>
      </Space.Compact block>
    </>
  );
};
```

```

        <Button onClick={() => setStatus('error')}>Status Error</Button>
      </Space.Compact>
      <br />
      <Steps current={current} percent={percent} status={status} items=
{items} />
      <Steps current={current} percent={percent} status={status}
size="small" items={items} />
      <Steps
        current={current}
        percent={percent}
        status={status}
        direction="vertical"
        items={items}
      />
      <Steps
        current={current}
        percent={percent}
        status={status}
        size="small"
        direction="vertical"
        items={items}
      />
    </>
  );
};

export default App;

```

Steps inside Steps

Debug

```

import React, { useState } from 'react';
import type { StepsProps } from 'antd';
import { Card, Radio, Steps } from 'antd';

const App: React.FC = () => {
  const [size, setSize] = useState<StepsProps['size']>('default');
  const description = 'This is a description.';
  const horizontalSteps = (
    <Card>
      <Steps
        size={size}
        items={[
          {
            title: 'Finished',

```



```

        description,
      },
      {
        title: 'In Progress',
        description,
      },
      {
        title: 'Waiting',
        description,
      },
    ],
  ]}
  />
</Card>
);

return (
  <>
    <Radio.Group
      style={{ marginBottom: 16 }}
      value={size}
      onChange={(e) => setSize(e.target.value)}
    >
      <Radio value="small">Small</Radio>
      <Radio value="default">Default</Radio>
    </Radio.Group>
    <Steps
      size={size}
      direction="vertical"
      items={[
        {
          title: 'Finished',
          description: horizontalSteps,
        },
        {
          title: 'In Progress',
          description,
        },
        {
          title: 'Waiting',
          description,
        },
      ]}
    />
  </>
);
};

```

```
export default App;
```

Inline Steps

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

const data = [
  {
    title: 'Ant Design Title 1',
    current: 0,
  },
  {
    title: 'Ant Design Title 2',
    current: 1,
    status: 'error',
  },
  {
    title: 'Ant Design Title 3',
    current: 2,
  },
  {
    title: 'Ant Design Title 4',
    current: 1,
  },
];

const items = [
  {
    title: 'Step 1',
    description: 'This is a Step 1.',
  },
  {
    title: 'Step 2',
    description: 'This is a Step 2.',
  },
  {
    title: 'Step 3',
    description: 'This is a Step 3.',
  },
];

const App: React.FC = () => (
  <div>
```

```

<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"
      />
      <Steps
        style={{ marginTop: 8 }}
        type="inline"
        current={item.current}
        status={item.status as StepsProps['status']}
        items={items}
      />
    </List.Item>
  )}
/>
</div>
);

export default App;

```

Wireframe

Debug

```

import React from 'react';
import { ConfigProvider, Steps } from 'antd';

const description = 'This is a description.';
const App: React.FC = () => (
  <ConfigProvider theme={{ token: { wireframe: true } }}>
    <Steps
      current={1}
      items={[
        {
          title: 'Finished',
          description,
        },
        {
          title: 'In Progress',

```

```

        description,
        subTitle: 'Left 00:00:08',
      },
      {
        title: 'Waiting',
        description,
      },
    ]}
  />
</ConfigProvider>
);

export default App;

```

Component Token

Debug

```

import React from 'react';
import { LoadingOutlined, SmileOutlined, SolutionOutlined, UserOutlined }
from '@ant-design/icons';
import { ConfigProvider, Steps } from 'antd';

const description = 'This is a description.';
const App: React.FC = () => (
  <ConfigProvider
    theme={{
      components: {
        Steps: {
          titleLineHeight: 20,
          customIconSize: 40,
          customIconTop: 0,
          customIconFontSize: 32,
          iconSize: 20,
          iconTop: 0, // magic for ui experience
          iconFontSize: 12,
          iconSizeSM: 16,
          dotSize: 20,
          dotCurrentSize: 24,
          navArrowColor: '#163CFF',
          navContentMaxWidth: 100,
          descriptionMaxWidth: 100,
        },
      },
    }}
  >

```

```
<Steps
  current={1}
  items={[
    {
      title: 'Finished',
      description,
    },
    {
      title: 'In Progress',
      description,
      subTitle: 'Left 00:00:08',
    },
    {
      title: 'Waiting',
      description,
    },
  ]}
/>
<Steps
  size="small"
  current={1}
  items={[
    {
      title: 'Finished',
      description,
    },
    {
      title: 'In Progress',
      description,
      subTitle: 'Left 00:00:08',
    },
    {
      title: 'Waiting',
      description,
    },
  ]}
/>
<Steps
  items={[
    {
      title: 'Login',
      status: 'finish',
      icon: <UserOutlined />,
    },
    {
      title: 'Verification',
```

```

        status: 'finish',
        icon: <SolutionOutlined />,
      },
      {
        title: 'Pay',
        status: 'process',
        icon: <LoadingOutlined />,
      },
      {
        title: 'Done',
        status: 'wait',
        icon: <SmileOutlined />,
      },
    ],
  ]}
/>
<Steps
  progressDot
  current={1}
  items={[
    {
      title: 'Finished',
      description: 'This is a description.',
    },
    {
      title: 'In Progress',
      description: 'This is a description.',
    },
    {
      title: 'Waiting',
      description: 'This is a description.',
    },
  ]}
/>
<Steps
  type="navigation"
  current={1}
  items={[
    {
      title: 'Step 1',
      subTitle: '00:00:05',
      status: 'finish',
      description: 'This is a description.',
    },
    {
      title: 'Step 2',
      subTitle: '00:01:02',

```

```

        status: 'process',
        description: 'This is a description.',
      },
      {
        title: 'Step 3',
        subTitle: 'waiting for longlong time',
        status: 'wait',
        description: 'This is a description.',
      },
    ]
  }
  />
</ConfigProvider>
);

export default App;

```

API

Common props ref: [Common props](#)

Steps

The whole of the step bar.

Property	Description	Type	Default	Version
className	Additional class to Steps	string	-	
current	To set the current step, counting from 0. You can overwrite this state by using status of Step	number	0	
direction	To specify the direction of the step bar, <code>horizontal</code> or <code>vertical</code>	string	<code>horizontal</code>	
initial	Set the initial step, counting from 0	number	0	
labelPlacement	Place title and description with <code>horizontal</code> or <code>vertical</code> direction	string	<code>horizontal</code>	
percent	Progress circle percentage of current step in process status	number	-	4.5.0

	(only works on basic Steps)			
progressDot	Steps with progress dot style, customize the progress dot by setting it to a function. labelPlacement will be vertical	boolean (iconDot, {index, status, title, description}) => ReactNode	false	
responsive	Change to vertical direction when screen width smaller than 532px	boolean	true	
size	To specify the size of the step bar, default and small are currently supported	string	default	
status	To specify the status of current step, can be set to one of the following values: wait process finish error	string	process	
type	Type of steps, can be set to one of the following values: default navigation inline	string	default	inline: 5.0
onChange	Trigger when Step is changed	(current) => void	-	
items	StepItem content	StepItem	[]	4.24.0

type="inline"

Property	Description	Type	Default	Version
className	Additional class to Steps	string	-	
current	To set the current step, counting from 0. You can overwrite this state by using status of Step	number	0	
initial	Set the initial step, counting from 0	number	0	
status	To specify the status of current step, can be set to one of the following values: wait process finish error	string	process	

onChange	Trigger when Step is changed	(current) => void	-	
items	StepItem content. not supported: icon subtitle	StepItem	[]	4.24.0

StepItem

A single step in the step bar.

Property	Description	Type	Default	Version
description	Description of the step, optional property	ReactNode	-	
disabled	Disable click	boolean	false	
icon	Icon of the step, optional property	ReactNode	-	
status	To specify the status. It will be automatically set by current of Steps if not configured. Optional values are: wait process finish error	string	wait	
subTitle	Subtitle of the step	ReactNode	-	
title	Title of the step	ReactNode	-	

Design Token