Timeline

Vertical display timeline.

When To Use

- When a series of information needs to be ordered by time (ascending or descending).
- When you need a timeline to make a visual connection.

After version 5.2.0, we provide a simpler usage <Timeline items={[...]} /> with better performance and potential of writing simpler code style in your applications. Meanwhile, we deprecated the old usage in browser console, we will remove it in antd 6.0.

```
// works when >=5.2.0, recommended ✅
const items = [{ children: 'sample', label: 'sample' }];
return <Timeline items={items} />;

// works when <5.2.0, deprecated when >=5.2.0 🙅🏻‍♀️
return (
  <Timeline onChange={onChange}>
    <Timeline.Item>Sample</Timeline.Item>
  </Timeline>
);
```

Examples

- Create a services site 2015-09-01
- Solve initial network problems 2015-09-01
- Technical testing 2015-09-01
- Network problems being solved 2015-09-01

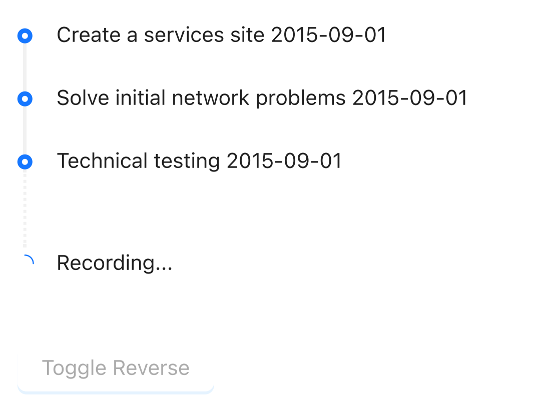
Basic

Basic timeline.

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

const App: React.FC = () => (
  <Timeline
    items={[
      {
        children: 'Create a services site 2015
      },
      {
        children: 'Solve initial network probl
      },
      {
        children: 'Technical testing 2015-09-0
      },
      {
        children: 'Network problems being sol
      },
    ]}
  />
);

export default App;
```



Create a services site 2015-09-01

Solve initial network problems 2015-09-01

Technical testing 2015-09-01

Recording...

Last node and Reversing

When the timeline is incomplete and ongoing, put a ghost node at last. Set `pending` as truthy value to enable displaying pending item. You can customize the pending content by passing a React Element. Meanwhile, `pendingDot={a React Element}` is used to customize the dot of the pending item. `reverse={true}` is used for reversing nodes.

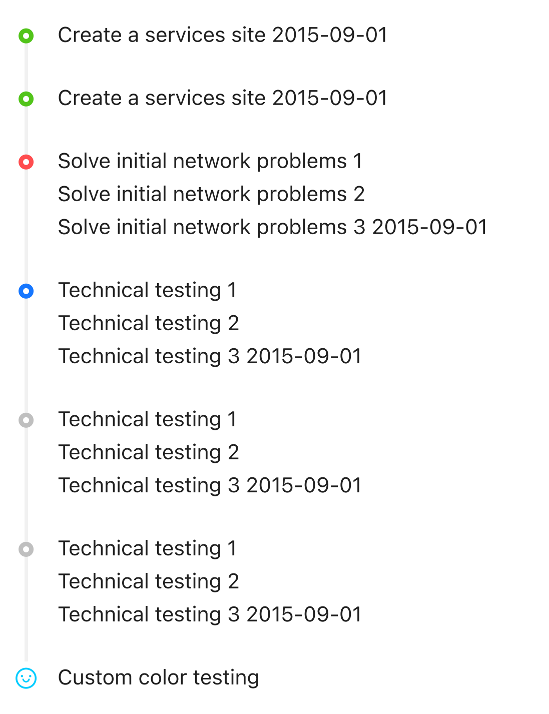
```
import React, { useState } from 'react';
import { Button, Timeline } from 'antd';

const App: React.FC = () => {
  const [reverse, setReverse] = useState(false

  const handleClick = () => {
    setReverse(!reverse);
  };

  return (
    <div>
      <Timeline
        pending="Recording..."
        reverse={reverse}
        items={[
          {
            children: 'Create a services site
          },
          {
            children: 'Solve initial network p
          },
          {
            children: 'Technical testing 2015-
          },
        ]}
      />
      <Button type="primary" style={{ marginT
        Toggle Reverse
      </Button>
    </div>
  );
};

export default App;
```



Create a services site 2015-09-01

Create a services site 2015-09-01

Solve initial network problems 1
Solve initial network problems 2
Solve initial network problems 3 2015-09-01

Technical testing 1
Technical testing 2
Technical testing 3 2015-09-01

Technical testing 1
Technical testing 2
Technical testing 3 2015-09-01

Technical testing 1
Technical testing 2
Technical testing 3 2015-09-01

Custom color testing

Color

Set the color of circles. `green` means completed or success status, `red` means warning or error, and `blue` means ongoing or other default status, `gray` for unfinished or disabled status.

```
import React from 'react';
import { SmileOutlined } from '@ant-design/ico
import { Timeline } from 'antd';

const App: React.FC = () => (
  <Timeline
    items={[
      {
        color: 'green',
        children: 'Create a services site 201
      },
      {
        color: 'green',
        children: 'Create a services site 201
      },
      {
        color: 'red',
        children: (
          <>
            <p>Solve initial network problems
            <p>Solve initial network problems
            <p>Solve initial network problems
          </>
        ),
      },
      {
        children: (
          <>
            <p>Technical testing 1</p>
            <p>Technical testing 2</p>
            <p>Technical testing 3 2015-09-01
          </>
        ),
      },
      {
        color: 'gray',
```

- Create a services site 2015-09-01
- Solve initial network problems 2015-09-01
- Technical testing 2015-09-01
- Network problems being solved 2015-09-01

Custom

Set a node as an icon or other custom element.

```
import React from 'react';
import { ClockCircleOutlined } from '@ant-desi
import { Timeline } from 'antd';

const App: React.FC = () => (
  <Timeline
    items={[
      {
        children: 'Create a services site 2015
      },
      {
        children: 'Solve initial network probl
      },
      {
        dot: <ClockCircleOutlined className="t
        color: 'red',
        children: 'Technical testing 2015-09-0
      },
      {
        children: 'Network problems being solv
      },
    ]}
  />
);

export default App;
```

Create a services site 2015-09-01

Solve initial network problems 2015-09-01

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.

Network problems being solved 2015-09-01

Create a services site 2015-09-01

Technical testing 2015-09-01

Alternate

Alternate timeline.

```
import React from 'react';
import { ClockCircleOutlined } from '@ant-desi
import { Timeline } from 'antd';

const App: React.FC = () => (
  <Timeline
    mode="alternate"
    items={[
      {
        children: 'Create a services site 2015
      },
      {
        children: 'Solve initial network probl
        color: 'green',
      },
      {
        dot: <ClockCircleOutlined style={{ for
        children: `Sed ut perspiciatis unde om
      },
      {
        color: 'red',
        children: 'Network problems being solv
      },
      {
        children: 'Create a services site 2015
      },
      {
        dot: <ClockCircleOutlined style={{ for
        children: 'Technical testing 2015-09-0
      },
    ]}
  />
);

export default App;
```

[✓ Left] Right Alternate

2015-09-01 ● Create a services

2015-09-01 09:12:11 ● Solve initial network problems

● Technical testing

2015-09-01 09:12:11 ● Network problems being solved

Label

Use `label` show time alone.

```
import React, { useState } from 'react';
import type { RadioChangeEvent } from 'antd';
import { Radio, Timeline } from 'antd';

const App: React.FC = () => {
  const [mode, setMode] = useState<'left' | 'r

  const onChange = (e: RadioChangeEvent) => {
    setMode(e.target.value);
  };

  return (
    <>
      <Radio.Group
        onChange={onChange}
        value={mode}
        style={{
          marginBottom: 20,
        }}
      >
        <Radio value="left">Left</Radio>
        <Radio value="right">Right</Radio>
        <Radio value="alternate">Alternate</Ra
      </Radio.Group>
      <Timeline
        mode={mode}
        items={[
          {
            label: '2015-09-01',
            children: 'Create a services',
          },
          {
            label: '2015-09-01 09:12:11',
            children: 'Solve initial network p
          },
          {
            children: 'Technical testing',
          },
          {
            label: '2015-09-01 09:12:11',
            children: 'Network problems being
          },
        ]}
      />
    </>
  );
};

export default App;
```

Create a services site 2015-09-01 ●

Solve initial network problems 2015-09-01 ●

Technical testing 2015-09-01 🕓

Network problems being solved 2015-09-01 ●

Right alternate

Right alternate timeline.

```
import React from 'react';
import { ClockCircleOutlined } from '@ant-desi
import { Timeline } from 'antd';

const App: React.FC = () => (
  <Timeline
    mode="right"
    items={[
      {
        children: 'Create a services site 2015
      },
      {
        children: 'Solve initial network probl
      },
      {
        dot: <ClockCircleOutlined style={{ for
        color: 'red',
        children: 'Technical testing 2015-09-0
      },
      {
        children: 'Network problems being solv
      },
    ]}
  />
);

export default App;
```

API

Timeline

Property	Description	Type	Default
mode	By sending `alternate` the timeline will distribute the nodes to the left and right	`left` | `alternate` | `right`	-
pending	Set the last ghost node's existence or its content	boolean | ReactNode	false
pendingDot	Set the dot of the last ghost node when pending is true	ReactNode	<LoadingOutlined />
reverse	Whether reverse nodes or not	boolean	false
items	Each node of timeline	Items[]	5.2.0

Items

Node of timeline.

Property	Description	Type	Default
color	Set the circle's color to `blue`, `red`, `green`, `gray` or other custom colors	string	`blue`
dot	Customize timeline dot	ReactNode	-
label	Set the label	ReactNode	-
children	Set the content	ReactNode	-
position	Customize node position	`left` | `right`	-

Design Token

▾ Global Token

Token Name	Description	Type	Default Value
colorBgContainer	Container background color, e.g: default button, input box, etc. Be sure not to confuse this with `colorBgElevated`.	`string`	⬜ #ffffff
colorError	Used to represent the visual elements of the operation failure, such as the error Button, error Result component, etc.	`string`	⬜ #ff4d4f
colorPrimary	Brand color is one of the most direct visual elements to reflect the characteristics and communication of the product. After you have selected the brand color, we will automatically generate a complete color palette and assign it effective design semantics.	`string`	⬜ #1677ff

Token Name	Description	Type	Default Value
colorSplit	Used as the color of separator, this color is the same as colorBorderSecondary but with transparency.	`string`	□ `rgba(5, 5, 5, 0.06)`
colorSuccess	Used to represent the token sequence of operation success, such as Result, Progress and other components will use these map tokens.	`string`	□ `#52c41a`
colorText	Default text color which comply with W3C standards, and this color is also the darkest neutral color.	`string`	□ `rgba(0, 0, 0, 0.88)`
colorTextDisabled	Control the color of text in disabled state.	`string`	□ `rgba(0, 0, 0, 0.25)`
controlHeightLG	LG component height	`number`	40
fontFamily	The font family of Ant Design prioritizes the default interface font of the system, and provides a set of alternative font libraries that are suitable for screen display to maintain the readability and readability of the font under different platforms and browsers, reflecting the friendly, stable and professional characteristics.	`string`	-apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, 'Helvetica Neue', Arial, 'Noto Sans', sans-serif, 'Apple Color Emoji', 'Segoe UI Emoji', 'Segoe UI Symbol', 'Noto Color Emoji'
fontSize	The most widely used font size in the design system, from which the text gradient will be derived.	`number`	14
fontSizeSM	Small font size	`number`	12
lineHeight	Line height of text.	`number`	1.5714285714285714
lineType	Border style of base components	`string`	solid
lineWidth	Border width of base components	`number`	1
lineWidthBold	The default line width of the outline class components, such as Button, Input, Select, etc.	`number`	2
margin	Control the margin of an element, with a medium size.	`number`	16
marginSM	Control the margin of an element, with a medium-small size.	`number`	12
marginXS	Control the margin of an element, with a small size.	`number`	8
marginXXS	Control the margin of an element, with the smallest size.	`number`	4
padding	Control the padding of the element.	`number`	16
paddingXXS	Control the extra extra small padding of the element.	`number`	4

Token Name	Description	Type	Default Value
wireframe	Used to change the visual effect of the component to wireframe, if you need to use the V4 effect, you need to enable the configuration item	`boolean`	false