When To Use

By clicking the input box, you can select a time from a popup panel.

Examples

Basic

```
import React from 'react';
import type { TimePickerProps } from 'antd';
import { TimePicker } from 'antd';
import dayjs from 'dayjs';
import customParseFormat from 'dayjs/plugin/customParseFormat';

dayjs.extend(customParseFormat);

const onChange: TimePickerProps['onChange'] = (time, timeString) => {
  console.log(time, timeString);
};

const App: React.FC = () => (
  <TimePicker onChange={onChange} defaultOpenValue={dayjs('00:00:00',
'HH:mm:ss')} />
);

export default App;
```

Under Control

```
import React, { useState } from 'react';
import { TimePicker } from 'antd';
import type { Dayjs } from 'dayjs';

const App: React.FC = () => {
  const [value, setValue] = useState<Dayjs | null>(null);

  const onChange = (time: Dayjs) => {
    setValue(time);
  };

  return <TimePicker value={value} onChange={onChange} />;
};

export default App;
```

Three Sizes

```
import React from 'react';
import { Space, TimePicker } from 'antd';
import dayjs from 'dayjs';

const App: React.FC = () => (
  <Space wrap>
    <TimePicker defaultValue={dayjs('12:08:23', 'HH:mm:ss')} size="large"
/>
    <TimePicker defaultValue={dayjs('12:08:23', 'HH:mm:ss')} />
    <TimePicker defaultValue={dayjs('12:08:23', 'HH:mm:ss')} size="small"
/>
  </Space>
);

export default App;
```

Need Confirm

v5.14.0

```
import React from 'react';
import type { TimePickerProps } from 'antd';
import { TimePicker } from 'antd';

const onChange: TimePickerProps['onChange'] = (time, timeString) => {
  console.log(time, timeString);
};

const App: React.FC = () => <TimePicker onChange={onChange} needConfirm />;

export default App;
```

disabled

```
import React from 'react';
import { TimePicker } from 'antd';
import dayjs from 'dayjs';
import customParseFormat from 'dayjs/plugin/customParseFormat';

dayjs.extend(customParseFormat);

const App: React.FC = () => <TimePicker defaultValue={dayjs('12:08:23',
'HH:mm:ss')} disabled />;
```

```
export default App;
```

Hour and minute

```
import React from 'react';
import { TimePicker } from 'antd';
import dayjs from 'dayjs';

const format = 'HH:mm';

const App: React.FC = () => <TimePicker defaultValue={dayjs('12:08',
format)} format={format} />;

export default App;
```

interval option

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

const App: React.FC = () => <TimePicker minuteStep={15} secondStep={10}
hourStep={1} />;

export default App;
```

Addon

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

const App: React.FC = () => {
  const [open, setOpen] = useState(false);

  return (
    <TimePicker
      open={open}
      onOpenChange={setOpen}
      renderExtraFooter={() => (
        <Button size="small" type="primary" onClick={() => setOpen(false)}>
          OK
        </Button>
      )}
    />
  );
};

export default App;
```

12 hours

```
import React from 'react';
import type { TimePickerProps } from 'antd';
import { Space, TimePicker } from 'antd';

const onChange: TimePickerProps['onChange'] = (time, timeString) => {
  console.log(time, timeString);
};

const App: React.FC = () => (
  <Space wrap>
    <TimePicker use12Hours onChange={onChange} />
    <TimePicker use12Hours format="h:mm:ss A" onChange={onChange} />
    <TimePicker use12Hours format="h:mm a" onChange={onChange} />
  </Space>
);

export default App;
```

Change on scroll

v5.14.0

```
import React from 'react';
import type { TimePickerProps } from 'antd';
import { TimePicker } from 'antd';
import dayjs from 'dayjs';
import customParseFormat from 'dayjs/plugin/customParseFormat';

dayjs.extend(customParseFormat);

const onChange: TimePickerProps['onChange'] = (time, timeString) => {
  console.log(time, timeString);
};

const App: React.FC = () => <TimePicker onChange={onChange} changeOnScroll
needConfirm={false} />;

export default App;
```

Colored Popup

Debug

```
import React from 'react';
import type { TimePickerProps } from 'antd';
import { TimePicker } from 'antd';
import dayjs from 'dayjs';
import customParseFormat from 'dayjs/plugin/customParseFormat';

dayjs.extend(customParseFormat);

const onChange: TimePickerProps['onChange'] = (time, timeString) => {
  console.log(time, timeString);
};

const App: React.FC = () => (
  <TimePicker
    onChange={onChange}
    defaultOpenValue={dayjs('00:00:00', 'HH:mm:ss')}
    popupClassName="myCustomClassName"
  />
);

export default App;
```

Time Range Picker

```
import React from 'react';
import { TimePicker } from 'antd';
import dayjs from 'dayjs';

const format = 'HH:mm:ss';

const App: React.FC = () => {
  const startTime = dayjs('12:08:23', 'HH:mm:ss');
  const endTime = dayjs('12:08:23', 'HH:mm:ss');

  return <TimePicker.RangePicker defaultValue={[startTime, endTime]}
format={format} />;
};

export default App;
```

Variants

v5.13.0

```
import React from 'react';
import { Flex, TimePicker } from 'antd';

const { RangePicker } = TimePicker;

const App: React.FC = () => (
  <Flex vertical gap={12}>
    <Flex gap={8}>
      <TimePicker placeholder="Outlined" />
      <RangePicker placeholder={['Outlined Start', 'Outlined End']} />
    </Flex>
    <Flex gap={8}>
      <TimePicker variant="filled" placeholder="Filled" />
      <RangePicker variant="filled" placeholder={['Filled Start', 'Filled
End']} />
    </Flex>
    <Flex gap={8}>
      <TimePicker variant="borderless" placeholder="Borderless" />
      <RangePicker variant="borderless" placeholder={['Borderless Start',
'Borderless End']} />
    </Flex>
    <Flex gap={8}>
      <TimePicker variant="underlined" placeholder="Underlined" />
      <RangePicker variant="underlined" placeholder={['Underlined Start',
'Underlined End']} />
    </Flex>
  </Flex>
);

export default App;
```

Status

```
import React from 'react';
import { Space, TimePicker } from 'antd';

const App: React.FC = () => (
  <Space direction="vertical">
    <TimePicker status="error" />
    <TimePicker status="warning" />
    <TimePicker.RangePicker status="error" />
    <TimePicker.RangePicker status="warning" />
  </Space>
);

export default App;
```

Prefix and Suffix

```
import React from 'react';
import { SmileOutlined } from '@ant-design/icons';
import { Space, TimePicker } from 'antd';
import type { TimePickerProps } from 'antd';
import dayjs from 'dayjs';
import customParseFormat from 'dayjs/plugin/customParseFormat';

dayjs.extend(customParseFormat);

const onChange: TimePickerProps['onChange'] = (time, timeString) => {
  console.log(time, timeString);
};

const App: React.FC = () => (
  <Space direction="vertical" size={12}>
    <TimePicker
      suffixIcon={<SmileOutlined />}
      onChange={onChange}
      defaultOpenValue={dayjs('00:00:00', 'HH:mm:ss')}
    />
    <TimePicker prefix={<SmileOutlined />} />
    <TimePicker.RangePicker prefix={<SmileOutlined />} />
  </Space>
);

export default App;
```

_InternalPanelDoNotUseOrYouWillBeFired

Debug

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

const { _InternalPanelDoNotUseOrYouWillBeFired: InternalTimePicker } =
TimePicker;

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

```
export default App;
```

API

Common props ref: Common props

```
import dayjs from 'dayjs';
import customParseFormat from 'dayjs/plugin/customParseFormat'

dayjs.extend(customParseFormat)

<TimePicker defaultValue={dayjs('13:30:56', 'HH:mm:ss')} />;
```

Property	Description	Type	Default	Version
allowClear	Customize clear icon	boolean | { clearIcon?: ReactNode }	true	5.8.0: Support object type
autoFocus	If get focus when component mounted	boolean	false	
cellRender	Custom rendering function for picker cells	(current: number, info: { originNode: React.ReactElement, today: dayjs, range?: 'start' | 'end', subType: 'hour' | 'minute' | 'second' | 'meridiem' }) => React.ReactNode	-	5.4.0
changeOnScroll	Trigger selection when scroll the column	boolean	false	5.14.0
className	The className of picker	string	-	
defaultValue	To set default time	dayjs	-	

disabled	Determine whether the TimePicker is disabled	boolean	false	
disabledTime	To specify the time that cannot be selected	[DisabledTime]()	-	4.19.0
format	To set the time format	string	`HH:mm:ss`	
getPopupContainer	To set the container of the floating layer, while the default is to create a div element in body	function(trigger)	-	
hideDisabledOptions	Whether hide the options that can not be selected	boolean	false	
hourStep	Interval between hours in picker	number	1	
inputReadOnly	Set the `readonly` attribute of the input tag (avoids virtual keyboard on touch devices)	boolean	false	
minuteStep	Interval between minutes in picker	number	1	
needConfirm	Need click confirm button to trigger value change	boolean	-	5.14.0

open	Whether to popup panel	boolean	false	
placeholder	Display when there's no value	string | [string, string]	`Select a time`	
placement	The position where the selection box pops up	`bottomLeft` `bottomRight` `topLeft` `topRight`	bottomLeft	
popupClassName	The className of panel	string	-	
popupStyle	The style of panel	CSSProperties	-	
prefix	The custom prefix	ReactNode	-	5.22.0
renderExtraFooter	Called from time picker panel to render some addon to its bottom	() => ReactNode	-	
secondStep	Interval between seconds in picker	number	1	
showNow	Whether to show `Now` button on panel	boolean	-	4.4.0
size	To determine the size of the input box, the height of `large` and `small`, are 40px and 24px respectively, while default size is 32px	`large` | `middle` | `small`	-	
status	Set validation status	'error' | 'warning' | 'success' | 'validating'	-	4.19.0

suffixIcon	The custom suffix icon	ReactNode	-	
use12Hours	Display as 12 hours format, with default format `h:mm:ss a`	boolean	false	
value	To set time	[dayjs]()	-	
variant	Variants of picker	`outlined` | `borderless` | `filled` | `underlined`	`outlined`	5.13.0 | `underlined`: 5.24.0
onCalendarChange	Callback function, can be executed when the start time or the end time of the range is changing. `info` argument is added in 4.4.0	function(dates: [dayjs, dayjs], dateStrings: [string, string], info: { range:`start`|`end` })	-	
onChange	A callback function, can be executed when the selected time is changing	function(time: dayjs, timeString: string): void	-	
onOpenChange	A callback function which will be called while panel opening/closing	(open: boolean) => void	-	

DisabledTime

```
type DisabledTime = (now: Dayjs) => {
  disabledHours?: () => number[];
  disabledMinutes?: (selectedHour: number) => number[];
  disabledSeconds?: (selectedHour: number, selectedMinute: number) =>
number[];
```

```
  disabledMilliseconds?: (
    selectedHour: number,
    selectedMinute: number,
    selectedSecond: number,
  ) => number[];
};
```

Note: `disabledMilliseconds` is added in `5.14.0`.

Methods

Name	Description	Version
blur()	Remove focus	
focus()	Get focus	

RangePicker

Same props from RangePicker of DatePicker. And includes additional props:

Property	Description	Type	Default	Version
disabledTime	To specify the time that cannot be selected	RangeDisabledTime	-	4.19.0
order	Order start and end time	boolean	true	4.1.0

RangeDisabledTime

```
type RangeDisabledTime = (
  now: Dayjs,
  type = 'start' | 'end',
) => {
  disabledHours?: () => number[];
  disabledMinutes?: (selectedHour: number) => number[];
  disabledSeconds?: (selectedHour: number, selectedMinute: number) =>
number[];
};
```

Design Token

FAQ

- How to use TimePicker with customize date library like dayjs