

何时使用 {#when-to-use}

当用户需要输入一个时间，可以点击标准输入框，弹出时间面板进行选择。

代码演示

基本

```
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;
```

受控组件

```
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;
```

三种大小

```
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;
```

选择确认

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;
```

禁用

```
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;
```

选择时分

```
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;
```

步长选项

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

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

export default App;
```

附加内容

```
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 小时制

```

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;

```

滚动即改变

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;

```

色付きポップアップ

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;
```

範囲選択器

```
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;
```

形態変体

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;
```

自定义状态

```
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;
```

前后缀

```
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

通用属性参考: [通用属性](#)

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

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

参数	说明	类型	默认值	版本
allowClear	自定义清除按钮	boolean { clearIcon?: ReactNode }	true	5.8.0: 支持对象类型
autoFocus	自动获取焦点	boolean	false	
cellRender	自定义单元格的内容	(current: number, info: { originNode: React.ReactNode, today: dayjs, range?: 'start' 'end', subType: 'hour' 'minute' 'second' 'meridiem' }) => React.ReactNode	-	5.4.0
changeOnScroll	在滚动时改变选择值	boolean	false	5.14.0
className	选择器类名	string	-	
defaultValue	默认时间	dayjs	-	
disabled	禁用全部操作	boolean	false	
disabledTime	不可选择的时间	DisabledTime	-	4.19.0
format	展示的时间格式	string	HH:mm:ss	
getPopupContainer	定义浮层的容器，默认	function(trigger)	-	

	为 body 上新建 div			
hideDisabledOptions	隐藏禁止选择的选项	boolean	false	
hourStep	小时选项间隔	number	1	
inputReadOnly	设置输入框为只读（避免在移动设备上打开虚拟键盘）	boolean	false	
minuteStep	分钟选项间隔	number	1	
needConfirm	是否需要确认按钮，为 false 时失去焦点即代表选择	boolean	-	5.14.0
open	面板是否打开	boolean	false	
placeholder	没有值的时候显示的内容	string [string, string]	请选择时间	
placement	选择框弹出的位置	bottomLeft bottomRight topLeft topRight	bottomLeft	
popupClassName	弹出层类名	string	-	
popupStyle	弹出层样式对象	object	-	
prefix	自定义前缀	ReactNode	-	5.22.0
renderExtraFooter	选择框底部显示自定义的内容	() => ReactNode	-	
secondStep	秒选项间隔	number	1	
showNow	面板是否显示“此刻”按钮	boolean	-	4.4.0

size	输入框大小, large 高度为 40px, small 为 24px, 默认是 32px	large middle small	-	
status	设置校验状态	'error' 'warning'	-	4.19.0
suffixIcon	自定义的选择框后缀图标	ReactNode	-	
use12Hours	使用 12 小时制, 为 true 时 format 默认为 h:mm:ss a	boolean	false	
value	当前时间	dayjs	-	
variant	形态变体	outlined borderless filled underlined	outlined	5.13.0 underlined: 5.24.0
onCalendarChange	待选日期发生变化的回调。info 参数自 4.4.0 添加	function(dates: [dayjs, dayjs], dateStrings: [string, string], info: { range: start end })	-	
onChange	时间发生变化的回调	function(time: dayjs, timeString: string): void	-	
onOpenChange	面板打开/关闭时的回调	(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[];
};
```

注意：disabledMilliseconds 为 5.14.0 新增。

方法

名称	描述	版本
blur()	移除焦点	
focus()	获取焦点	

RangePicker

属性与 DatePicker 的 [RangePicker](#) 相同。还包含以下属性：

参数	说明	类型	默认值	版本
disabledTime	不可选择的时间	RangeDisabledTime	-	4.19.0
order	始末时间是否自动排序	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[];
};
```

主题变量（Design Token）

FAQ

- [如何在 TimePicker 中使用自定义日期库（如 Moment.js）](#)