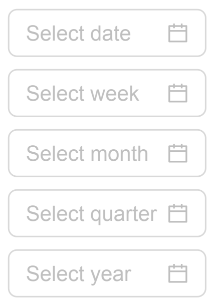
DatePicker

To select or input a date.

When To Use

By clicking the input box, you can select a date from a popup calendar.

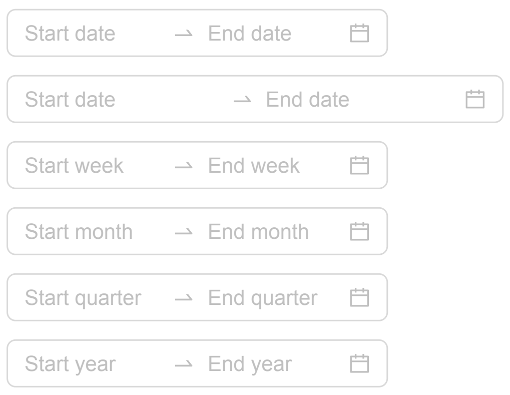
Examples



Basic

Basic use case. Users can select or input a date in panel.

```
import type { DatePickerProps } from 'antd';
import { DatePicker, Space } from 'antd';
import React from 'react';

const onChange: DatePickerProps['onChange'] =
  console.log(date, dateString);
};

const App: React.FC = () => (
  <Space direction="vertical">
    <DatePicker onChange={onChange} />
    <DatePicker onChange={onChange} picker="we
    <DatePicker onChange={onChange} picker="mo
    <DatePicker onChange={onChange} picker="qu
    <DatePicker onChange={onChange} picker="ye
  </Space>
);

export default App;
```



Range Picker

Set range picker type by `picker` prop.

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

const { RangePicker } = DatePicker;

const App: React.FC = () => (
  <Space direction="vertical" size={12}>
    <RangePicker />
    <RangePicker showTime />
    <RangePicker picker="week" />
    <RangePicker picker="month" />
    <RangePicker picker="quarter" />
    <RangePicker picker="year" />
  </Space>
);

export default App;
```

Time Select time

Switchable picker

Switch in different types of pickers by Select.

```
import React, { useState } from 'react';
import type { DatePickerProps, TimePickerProps
import { DatePicker, Select, Space, TimePicker

const { Option } = Select;

type PickerType = 'time' | 'date';

const PickerWithType = ({
  type,
  onChange,
}: {
  type: PickerType;
  onChange: TimePickerProps['onChange'] | Date
}) => {
  if (type === 'time') return <TimePicker onCh
  if (type === 'date') return <DatePicker onCh
  return <DatePicker picker={type} onChange={o
};

const App: React.FC = () => {
  const [type, setType] = useState<PickerType>

  return (
    <Space>
      <Select value={type} onChange={setType}>
        <Option value="time">Time</Option>
        <Option value="date">Date</Option>
        <Option value="week">Week</Option>
        <Option value="month">Month</Option>
        <Option value="quarter">Quarter</Optio
        <Option value="year">Year</Option>
      </Select>
      <PickerWithType type={type} onChange={(v
    </Space>
  );
};

export default App;
```

Select date

Select date

Select month

Select week

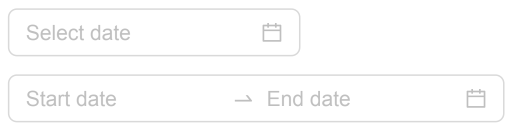
Start date → End date

Select date

Date Format

We can set the date format by `format`. When `format` is an array, the input box can be entered in any of the valid formats of the array.

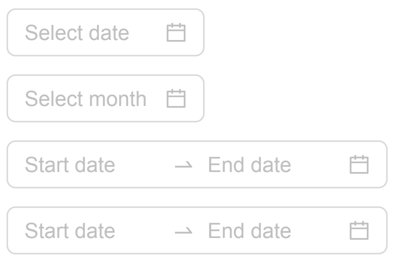
```
import React from 'react';
import type { DatePickerProps } from 'antd';
import { DatePicker, Space } from 'antd';
import dayjs from 'dayjs';
import customParseFormat from 'dayjs/plugin/cu

dayjs.extend(customParseFormat);

const { RangePicker } = DatePicker;

const dateFormat = 'YYYY/MM/DD';
const weekFormat = 'MM/DD';
const monthFormat = 'YYYY/MM';

/** Manually entering any of the following for
const dateFormatList = ['DD/MM/YYYY', 'DD/MM/Y

const customFormat: DatePickerProps['format']
  `custom format: ${value.format(dateFormat)}`

const customWeekStartEndFormat: DatePickerProp
  `${dayjs(value).startOf('week').format(weekF
    .endOf('week')
    .format(weekFormat)}`;

const App: React.FC = () => (
  <Space direction="vertical" size={12}>
    <DatePicker defaultValue={dayjs('2015/01/0
    <DatePicker defaultValue={dayjs('01/01/201
    <DatePicker defaultValue={dayjs('2015/01',
    <DatePicker defaultValue={dayjs()} format=
    <RangePicker
      defaultValue={[dayjs('2015/01/01', dateF
      format={dateFormat}
    />
    <DatePicker defaultValue={dayjs('2015/01/0
  </Space>
);

export default App;
```



Choose Time

This property provide an additional time selection. When `showTime` is an Object, its properties will be passed on to built-in `TimePicker`.

```
import React from 'react';
import { DatePicker, Space } from 'antd';
import type { DatePickerProps, RangePickerProp

const { RangePicker } = DatePicker;

const onChange = (
  value: DatePickerProps['value'] | RangePicke
  dateString: [string, string] | string,
) => {
  console.log('Selected Time: ', value);
  console.log('Formatted Selected Time: ', dat
};

const onOk = (value: DatePickerProps['value']
  console.log('onOk: ', value);
};

const App: React.FC = () => (
  <Space direction="vertical" size={12}>
    <DatePicker showTime onChange={onChange} c
    <RangePicker
      showTime={{ format: 'HH:mm' }}
      format="YYYY-MM-DD HH:mm"
      onChange={onChange}
      onOk={onOk}
    />
  </Space>
);

export default App;
```



Disabled

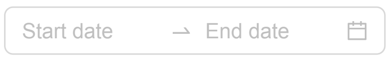
A disabled state of the `DatePicker`. You can also set as array to disable one of input.

```
import React from 'react';
import { DatePicker, Space } from 'antd';
import dayjs from 'dayjs';
import customParseFormat from 'dayjs/plugin/cu

dayjs.extend(customParseFormat);

const { RangePicker } = DatePicker;

const dateFormat = 'YYYY-MM-DD';

const App: React.FC = () => (
  <Space direction="vertical" size={12}>
    <DatePicker defaultValue={dayjs('2015-06-0
    <DatePicker picker="month" defaultValue={c
    <RangePicker
      defaultValue={[dayjs('2015-06-06', dateF
      disabled
    />
    <RangePicker
      defaultValue={[dayjs('2019-09-03', dateF
      disabled={[false, true]}
    />
  </Space>
);

export default App;
```

Disabled Date & Time

Disabled part of dates and time by `disabledDate` and `disabledTime` respectively, and `disabledTime` only works with `showTime`.

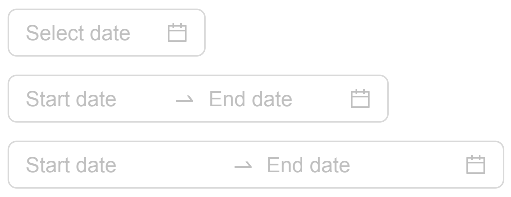
```
import React from 'react';
import { DatePicker, Space } from 'antd';
import type { RangePickerProps } from 'antd/es
import dayjs from 'dayjs';
import customParseFormat from 'dayjs/plugin/cu

dayjs.extend(customParseFormat);

const { RangePicker } = DatePicker;

const range = (start: number, end: number) =>
  const result = [];
  for (let i = start; i < end; i++) {
    result.push(i);
  }
  return result;
};

// eslint-disable-next-line arrow-body-style
const disabledDate: RangePickerProps['disablec
  // Can not select days before today and toda
  return current && current < dayjs().endOf('c
};

const disabledDateTime = () => ({
  disabledHours: () => range(0, 24).splice(4,
  disabledMinutes: () => range(30, 60),
  disabledSeconds: () => [55, 56],
});

const disabledRangeTime: RangePickerProps['dis
  if (type === 'start') {
    return {
      disabledHours: () => range(0, 60).splice
      disabledMinutes: () => range(30, 60),
      disabledSeconds: () => [55, 56],
    };
  }
  return {
    disabledHours: () => range(0, 60).splice(2
    disabledMinutes: () => range(0, 31),
    disabledSeconds: () => [55, 56],
  };
};

const App: React.FC = () => (
  <Space direction="vertical" size={12}>
    <DatePicker
      format="YYYY-MM-DD HH:mm:ss"
      disabledDate={disabledDate}
      disabledTime={disabledDateTime}
      showTime={{ defaultValue: dayjs('00:00:0
    />
    <DatePicker picker="month" disabledDate={d
```



Select range dates in 7 days

A example shows how to select a dynamic range by using `onCalendarChange` and `disabledDate`.

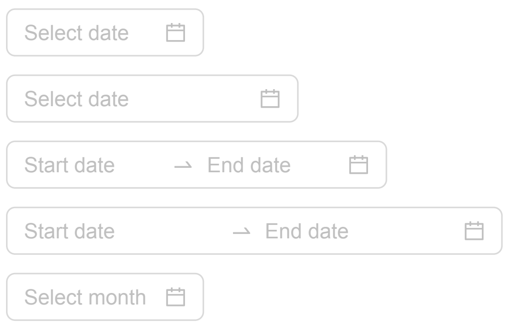
```
import React, { useState } from 'react';
import { DatePicker } from 'antd';
import type { Dayjs } from 'dayjs';

const { RangePicker } = DatePicker;

type RangeValue = [Dayjs | null, Dayjs | null]

const App: React.FC = () => {
  const [dates, setDates] = useState<RangeValu
  const [value, setValue] = useState<RangeValu

  const disabledDate = (current: Dayjs) => {
    if (!dates) {
      return false;
    }
    const tooLate = dates[0] && current.diff(c
    const tooEarly = dates[1] && dates[1].diff
    return !!tooEarly || !!tooLate;
  };

  const onOpenChange = (open: boolean) => {
    if (open) {
      setDates([null, null]);
    } else {
      setDates(null);
    }
  };

  return (
    <RangePicker
      value={dates || value}
      disabledDate={disabledDate}
      onCalendarChange={(val) => setDates(val)
      onChange={(val) => setValue(val)}
      onOpenChange={onOpenChange}
    />
  );
};

export default App;
```



Preset Ranges

We can set preset ranges to RangePicker to improve user experience.

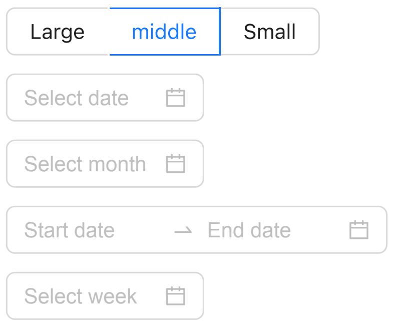
```
import React from 'react';
import { DatePicker, Space } from 'antd';
import dayjs from 'dayjs';
import type { Dayjs } from 'dayjs';

const { RangePicker } = DatePicker;

const onChange = (date: Dayjs) => {
  if (date) {
    console.log('Date: ', date);
  } else {
    console.log('Clear');
  }
};
const onRangeChange = (dates: null | (Dayjs |
  if (dates) {
    console.log('From: ', dates[0], ', to: ',
    console.log('From: ', dateStrings[0], ', t
  } else {
    console.log('Clear');
  }
};

const rangePresets: {
  label: string;
  value: [Dayjs, Dayjs];
}[] = [
  { label: 'Last 7 Days', value: [dayjs().add(
  { label: 'Last 14 Days', value: [dayjs().adc
  { label: 'Last 30 Days', value: [dayjs().adc
  { label: 'Last 90 Days', value: [dayjs().adc
];

const App: React.FC = () => (
  <Space direction="vertical" size={12}>
    <DatePicker
      presets={[
        { label: 'Yesterday', value: dayjs().c
        { label: 'Last Week', value: dayjs().c
        { label: 'Last Month', value: dayjs().
      ]}
      onChange={onChange}
    />
    <RangePicker presets={rangePresets} onChar
    <RangePicker
      presets={rangePresets}
      showTime
      format="YYYY/MM/DD HH:mm:ss"
      onChange={onRangeChange}
    />
  </Space>
);

export default App;
```



Extra Footer

Render extra footer in panel for customized requirements.

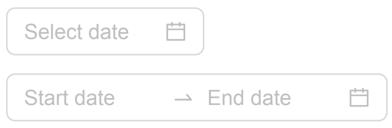
```
import React from 'react';
import { DatePicker, Space } from 'antd';

const { RangePicker } = DatePicker;

const App: React.FC = () => (
  <Space direction="vertical" size={12}>
    <DatePicker renderExtraFooter={() => 'extr
    <DatePicker renderExtraFooter={() => 'extr
    <RangePicker renderExtraFooter={() => 'ext
    <RangePicker renderExtraFooter={() => 'ext
    <DatePicker renderExtraFooter={() => 'extr
  </Space>
);

export default App;
```



Three Sizes

The input box comes in three sizes. `middle` will be used if `size` is omitted.

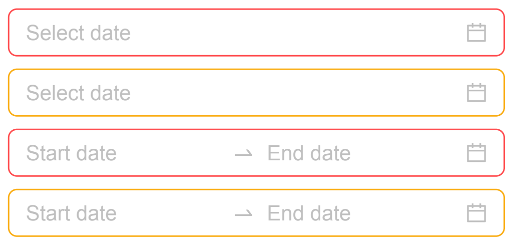
```
import React, { useState } from 'react';
import type { RadioChangeEvent } from 'antd';
import { DatePicker, Radio, Space } from 'antc
import type { SizeType } from 'antd/es/config-

const { RangePicker } = DatePicker;

const App: React.FC = () => {
  const [size, setSize] = useState<SizeType>('

  const handleSizeChange = (e: RadioChangeEver
    setSize(e.target.value);
  };

  return (
    <Space direction="vertical" size={12}>
      <Radio.Group value={size} onChange={hanc
        <Radio.Button value="large">Large</Rac
        <Radio.Button value="middle">middle</F
        <Radio.Button value="small">Small</Rac
      </Radio.Group>
      <DatePicker size={size} />
      <DatePicker size={size} picker="month" /
      <RangePicker size={size} />
      <DatePicker size={size} picker="week" />
    </Space>
  );
};

export default App;
```



Customized Cell Rendering

We can customize the rendering of the cells in the calendar by providing a `cellRender` function to `DatePicker`.

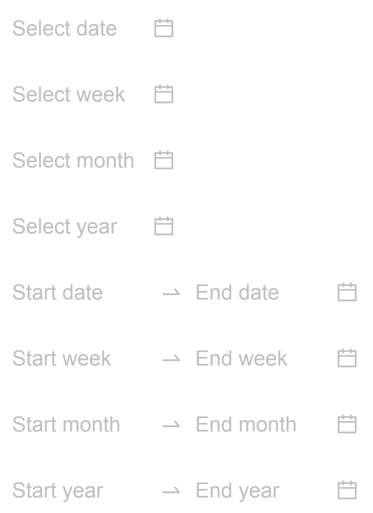
```
import { DatePicker, Space } from 'antd';
import React from 'react';

const { RangePicker } = DatePicker;

const App: React.FC = () => (
  <Space direction="vertical" size={12}>
    <DatePicker
      cellRender={(current) => {
        const style: React.CSSProperties = {};
        if (current.date() === 1) {
          style.border = '1px solid #1890ff';
          style.borderRadius = '50%';
        }
        return (
          <div className="ant-picker-cell-inne
            {current.date()}
          </div>
        );
      }}
    />
    <RangePicker
      cellRender={(current) => {
        const style: React.CSSProperties = {};
        if (current.date() === 1) {
          style.border = '1px solid #1890ff';
          style.borderRadius = '50%';
        }
        return (
          <div className="ant-picker-cell-inne
            {current.date()}
          </div>
        );
      }}
    />
  </Space>
);

export default App;
```



Status

Add status to DatePicker with `status`, which could be `error` or `warning`.

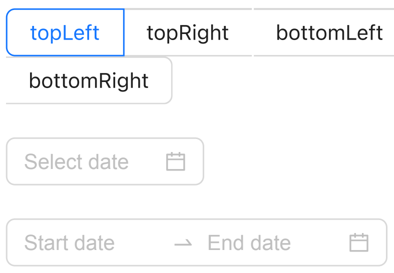
```
import React from 'react';
import { DatePicker, Space } from 'antd';

const App: React.FC = () => (
  <Space direction="vertical" style={{ width:
    <DatePicker status="error" style={{ width:
    <DatePicker status="warning" style={{ widt
    <DatePicker.RangePicker status="error" sty
    <DatePicker.RangePicker status="warning" s
  </Space>
);

export default App;
```



Bordered-less

Bordered-less style component.

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

const { RangePicker } = DatePicker;

const App: React.FC = () => (
  <Space direction="vertical" size={12}>
    <DatePicker bordered={false} />
    <DatePicker picker="week" bordered={false}
    <DatePicker picker="month" bordered={false
    <DatePicker picker="year" bordered={false}
    <RangePicker bordered={false} />
    <RangePicker picker="week" bordered={false
    <RangePicker picker="month" bordered={fals
    <RangePicker picker="year" bordered={false
  </Space>
);

export default App;
```



Placement

You can manually specify the position of the popup via `placement`.

```
import React, { useState } from 'react';
import type { DatePickerProps, RadioChangeEver
import { DatePicker, Radio } from 'antd';

const { RangePicker } = DatePicker;

const App: React.FC = () => {
  const [placement, SetPlacement] = useState<[

  const placementChange = (e: RadioChangeEvent
    SetPlacement(e.target.value);
  };

  return (
    <>
      <Radio.Group value={placement} onChange=
        <Radio.Button value="topLeft">topLeft<
        <Radio.Button value="topRight">topRigh
        <Radio.Button value="bottomLeft">bottc
        <Radio.Button value="bottomRight">bott
      </Radio.Group>
      <br />
      <br />
      <DatePicker placement={placement} />
      <br />
      <br />
      <RangePicker placement={placement} />
    </>
  );
};

export default App;
```

API

There are five kinds of picker:

- DatePicker
- DatePicker[picker="month"]
- DatePicker[picker="week"]
- DatePicker[picker="year"]
- DatePicker[picker="quarter"] (Added in 4.1.0)
- RangePicker

Localization

The default locale is en-US, if you need to use other languages, recommend to use internationalized components provided by us at the entrance. Look at: ConfigProvider.

If there are special needs (only modifying single component language), Please use the property: local. Example: default.

```
import 'dayjs/locale/zh-cn';
import locale from 'antd/es/date-picker/locale/zh_CN';

<DatePicker locale={locale} />;


// The default locale is en-US, if you want to use other locale, just set locale in entry file gl
import dayjs from 'dayjs';
import 'dayjs/locale/zh-cn';
import locale from 'antd/locale/zh_CN';

<ConfigProvider locale={locale}>
  <DatePicker defaultValue={dayjs('2015-01-01', 'YYYY-MM-DD')} />
</ConfigProvider>;
```

Common API

The following APIs are shared by DatePicker, RangePicker.

Property	Description	Type	Default
allowClear	Whether to show clear button	boolean	true
autoFocus	If get focus when component mounted	boolean	false
bordered	Whether has border style	boolean	true
className	The picker className	string	-
dateRender	Custom rendering function for date cells, >= 5.4.0 use `cellRender` instead.	function(currentDate: dayjs, today: dayjs) => React.ReactNode	-
cellRender	Custom rendering function for picker cells	function(current: dayjs, today: dayjs, info: { originNode: React.ReactElement,today: DateType, range?: 'start' | 'end', type: PanelMode, locale?: Locale, subType?: 'hour' | 'minute' | 'second' | 'meridiem' }) => React.ReactNode	-
disabled	Determine whether the DatePicker is disabled	boolean	false
disabledDate	Specify the date that cannot be	(currentDate: dayjs) => boolean	-

Property	Description	Type	Default
	selected		
format	To set the date format, support multi-format matching when it is an array, display the first one shall prevail. refer to dayjs#format. for example: Custom Format	formatType	rc-picker
popupClassName	To customize the className of the popup calendar	string	-
getPopupContainer	To set the container of the floating layer, while the default is to create a `div` element in `body`	function(trigger)	-
inputReadOnly	Set the `readonly` attribute of the input tag (avoids virtual keyboard on touch devices)	boolean	false
locale	Localization configuration	object	default
mode	The picker panel mode (Cannot select year or month anymore?)	`time` | `date` | `month` | `year` | `decade`	-
nextIcon	The custom next icon	ReactNode	-
open	The open state of picker	boolean	-
panelRender	Customize panel render	(panelNode) => ReactNode	-
picker	Set picker type	`date` | `week` | `month` | `quarter` | `year`	`date`
placeholder	The placeholder of date input	string | [string,string]	-
placement	The position where the selection box pops up	`bottomLeft` `bottomRight` `topLeft` `topRight`	bottomLeft
popupStyle	To customize the style of the popup calendar	CSSProperties	{}
presets	The preset ranges for quick selection	{ label: React.ReactNode, value: dayjs }[]	-

Property	Description	Type	Default
prevIcon	The custom prev icon	ReactNode	-
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'	-
style	To customize the style of the input box	CSSProperties	{}
suffixIcon	The custom suffix icon	ReactNode	-
superNextIcon	The custom super next icon	ReactNode	-
superPrevIcon	The custom super prev icon	ReactNode	-
onOpenChange	Callback function, can be executed whether the popup calendar is popped up or closed	function(open)	-
onPanelChange	Callback when picker panel mode is changed	function(value, mode)	-

Common Methods

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

DatePicker

Property	Description	Type	Default	Vers
defaultPickerValue	To set default picker date	dayjs	-	
defaultValue	To set default date, if start time or end time is null or undefined, the	dayjs	-	

Property	Description	Type	Default	Vers
	date range will be an open interval			
disabledTime	To specify the time that cannot be selected	function(date)	-	
format	To set the date format. refer to dayjs#format	formatType	`YYYY-MM-DD`	
renderExtraFooter	Render extra footer in panel	(mode) => React.ReactNode	-	
showNow	Whether to show 'Now' button on panel when `showTime` is set	boolean	-	4.4.
showTime	To provide an additional time selection	object | boolean	TimePicker Options	
showTime.defaultValue	To set default time of selected date, demo	dayjs	dayjs()	
showToday	Whether to show `Today` button	boolean	true	
value	To set date	dayjs	-	
onChange	Callback function, can be executed when the selected time is changing	function(date: dayjs, dateString: string)	-	
onOk	Callback when click ok button	function()	-	
onPanelChange	Callback function for panel changing	function(value, mode)	-	

DatePicker[picker=year]

Property	Description	Type	Default	Version
defaultPickerValue	To set default picker date	dayjs	-	
defaultValue	To set default date	dayjs	-	
format	To set the date format. refer to dayjs#format	formatType	`YYYY`	
renderExtraFooter	Render extra footer in panel	() => React.ReactNode	-	
value	To set date	dayjs	-	

Property	Description	Type	Default	Version
onChange	Callback function, can be executed when the selected time is changing	function(date: dayjs, dateString: string)	-	

DatePicker[picker=quarter]

Added in `4.1.0`.

Property	Description	Type	Default	Version
defaultPickerValue	To set default picker date	dayjs	-	
defaultValue	To set default date	dayjs	-	
format	To set the date format. refer to dayjs#format	formatType	`YYYY-\QQ`	
renderExtraFooter	Render extra footer in panel	() => React.ReactNode	-	
value	To set date	dayjs	-	
onChange	Callback function, can be executed when the selected time is changing	function(date: dayjs, dateString: string)	-	

DatePicker[picker=month]

Property	Description	Type	Default	Version
defaultPickerValue	To set default picker date	dayjs	-	
defaultValue	To set default date	dayjs	-	
format	To set the date format. refer to dayjs#format	formatType	`YYYY-MM`	
renderExtraFooter	Render extra footer in panel	() => React.ReactNode	-	
value	To set date	dayjs	-	
onChange	Callback function, can be executed when the selected time is changing	function(date: dayjs, dateString: string)	-	

DatePicker[picker=week]

Property	Description	Type	Default	Version
defaultPickerValue	To set default picker date	dayjs	-	
defaultValue	To set default date	dayjs	-	
format	To set the date format. refer to dayjs#format	formatType	`YYYY-wo`	
renderExtraFooter	Render extra footer in panel	(mode) => React.ReactNode	-	
value	To set date	dayjs	-	
onChange	Callback function, can be executed when the selected time is changing	function(date: dayjs, dateString: string)	-	

RangePicker

Property	Description	Type	Default
allowEmpty	Allow start or end input leave empty	[boolean, boolean]	[false, fa
dateRender	Custom rendering function for date cells, >= 5.4.0 use `cellRender` instead.	function(currentDate: dayjs, today: dayjs) => React.ReactNode	-
cellRender	Custom rendering function for picker cells	function(current: dayjs, today: dayjs, info: { originNode: React.ReactElement,today: DateType, range?: 'start' | 'end', type: PanelMode, locale?: Locale, subType?: 'hour' | 'minute' | 'second' | 'meridiem' }) => React.ReactNode	-
defaultPickerValue	To set default picker date	[dayjs, dayjs]	-
defaultValue	To set default date	[dayjs, dayjs]	-
disabled	If disable start or end	[boolean, boolean]	-
disabledTime	To specify the time that cannot be selected	function(date: dayjs, partial: `start` | `end`)	-
format	To set the date format. refer to dayjs#format	formatType	`YYYY-MM-DD HH:mm:ss`
presets	The preset ranges for quick selection	{ label: React.ReactNode, value: dayjs[] }[]	-

Property	Description	Type	Default
renderExtraFooter	Render extra footer in panel	() => React.ReactNode	-
separator	Set separator between inputs	React.ReactNode	`<SwapRight />`
showTime	To provide an additional time selection	object | boolean	TimePicke Options
showTime.defaultValue	To set default time of selected date, demo	dayjs[]	[dayjs(), dayjs()]
value	To set date	[dayjs, dayjs]	-
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	Callback function, can be executed when the selected time is changing	function(dates: [dayjs, dayjs], dateStrings: [string, string])	-

formatType

```
import type { Dayjs } from 'dayjs';

type Generic = string;
type GenericFn = (value: Dayjs) => string;

export type FormatType = Generic | GenericFn | Array<Generic | GenericFn>;
```

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
colorBgContainerDisabled	Control the background color of container in disabled state.	`string`	⬜ rgba(0, 0, 0, 0.04)
colorBgElevated	Container background color of the popup layer, in dark mode the color value of this token will be a little brighter than `colorBgContainer`. E.g: modal, pop-up, menu, etc.	`string`	⬜ #ffffff

Token Name	Description	Type	Default Value
colorBorder	Default border color, used to separate different elements, such as: form separator, card separator, etc.	`string`	`#d9d9d9`
colorError	Used to represent the visual elements of the operation failure, such as the error Button, error Result component, etc.	`string`	`#ff4d4f`
colorErrorOutline	Control the outline color of input component in error state.	`string`	`rgba(255, 38, 5, 0.06)`
colorIcon	Weak action. Such as `allowClear` or Alert close button	`string`	`rgba(0, 0, 0, 0.45)`
colorIconHover	Weak action hover color. Such as `allowClear` or Alert close button	`string`	`rgba(0, 0, 0, 0.88)`
colorLink	Control the color of hyperlink.	`string`	`#1677ff`
colorLinkActive	Control the color of hyperlink when clicked.	`string`	`#0958d9`
colorLinkHover	Control the color of hyperlink when hovering.	`string`	`#69b1ff`
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`
colorPrimaryBorder	The stroke color under the main color gradient, used on the stroke of components such as Slider.	`string`	`#91caff`
colorPrimaryHover	Hover state under the main color gradient.	`string`	`#4096ff`
colorSplit	Used as the color of separator, this color is the same as colorBorderSecondary but with transparency.	`string`	`rgba(5, 5, 5, 0.06)`
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)`
colorTextDescription	Control the font color of text description.	`string`	`rgba(0, 0, 0, 0.45)`
colorTextDisabled	Control the color of text in disabled state.	`string`	`rgba(0, 0, 0, 0.25)`
colorTextHeading	Control the font color of heading.	`string`	`rgba(0, 0, 0, 0.88)`
colorTextLightSolid	Control the highlight color of text with background color, such as the text in	`string`	`#fff`

Token Name	Description	Type	Default Value
	Primary Button components.		
colorTextPlaceholder	Control the color of placeholder text.	`string`	□ `rgba(0, 0, 0, 0.25)`
colorWarning	Used to represent the warning map token, such as Notification, Alert, etc. Alert or Control component(like Input) will use these map tokens.	`string`	□ `#faad14`
colorWarningOutline	Control the outline color of input component in warning state.	`string`	□ `rgba(255, 215, 5, 0.1)`
borderRadius	Border radius of base components	`number`	6
borderRadiusLG	LG size border radius, used in some large border radius components, such as Card, Modal and other components.	`number`	8
borderRadiusOuter	Outer border radius	`number`	4
borderRadiusSM	SM size border radius, used in small size components, such as Button, Input, Select and other input components in small size	`number`	4
borderRadiusXS	XS size border radius, used in some small border radius components, such as Segmented, Arrow and other components with small border radius.	`number`	2
boxShadowSecondary	Control the secondary box shadow style of an element.	`string`	0 6px 16px 0 rgba(0, 0, 0, 0.08), 0 3px 6px -4px rgba(0, 0, 0, 0.12), 0 9px 28px 8px rgba(0, 0, 0, 0.05)
controlHeight	The height of the basic controls such as buttons and input boxes in Ant Design	`number`	32
controlHeightLG	LG component height	`number`	40
controlHeightSM	SM component height	`number`	24
controlItemBgActive	Control the background color of control component item when active.	`string`	□ `#e6f4ff`
controlItemBgHover	Control the background color of control component item when hovering.	`string`	□ `rgba(0, 0, 0, 0.04)`
controlOutline	Control the outline color of input component.	`string`	□ `rgba(5, 145, 255, 0.1)`
controlOutlineWidth	Control the outline width of input component.	`number`	2
controlPaddingHorizontal	Control the horizontal padding of an element.	`number`	12
controlPaddingHorizontalSM	Control the horizontal padding of an element with a small-medium size.	`number`	8

Token Name	Description	Type	Default Value
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
fontSizeLG	Large font size	`number`	16
fontWeightStrong	Control the font weight of heading components (such as h1, h2, h3) or selected item.	`number`	600
lineHeight	Line height of text.	`number`	1.5714285714285714
lineHeightLG	Line height of large text.	`number`	1.5
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
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
motionDurationMid	Motion speed, medium speed. Used for medium element animation interaction.	`string`	0.2s
motionDurationSlow	Motion speed, slow speed. Used for large element animation interaction.	`string`	0.3s
motionEaseInOutCirc	Preset motion curve.	`string`	cubic-bezier(0.78, 0.14, 0.15, 0.86)
motionEaseInQuint	Preset motion curve.	`string`	cubic-bezier(0.755, 0.05, 0.855, 0.06)
motionEaseOutCirc	Preset motion curve.	`string`	cubic-bezier(0.08, 0.82, 0.17, 1)
motionEaseOutQuint	Preset motion curve.	`string`	cubic-bezier(0.23, 1, 0.32, 1)
padding	Control the padding of the element.	`number`	16
paddingSM	Control the small padding of the element.	`number`	12

Token Name	Description	Type	Default Value
paddingXS	Control the extra small padding of the element.	`number`	8
paddingXXS	Control the extra extra small padding of the element.	`number`	4
sizePopupArrow	The size of the component arrow	`number`	16
zIndexPopupBase	Base zIndex of component like FloatButton, Affix which can be cover by large popup	`number`	1000

FAQ

When set mode to DatePicker/RangePicker, cannot select year or month anymore?

Please refer FAQ

How to use DatePicker with customize date library like dayjs?

Please refer Use custom date library

Why config dayjs.locale globally not work?

DatePicker default set `locale` as `en` in v4. You can config DatePicker `locale` prop or ConfigProvider `locale` prop instead.

Date-related components locale is not working?

See FAQ Date-related-components-locale-is-not-working?

How to modify start day of week?

Please use correct language (#5605), or update dayjs `locale` config:

- Example: https://codesandbox.io/s/dayjs-day-of-week-x9tuj2?file=/demo.tsx

```
import dayjs from 'dayjs';
import 'dayjs/locale/zh-cn';
import 'dayjs/plugin/updateLocale';

dayjs.updateLocale('zh-cn', {
  weekStart: 0,
});
```

Why origin panel don't switch when using `panelRender`?

When you change the layout of nodes by `panelRender`, React will unmount and re-mount it which reset the component state. You should keep the layout stable. Please ref #27263 for more info.