

You might want to replace Moment.js with another date library (**Ant design currently supports [dayjs](#) and [date-fns](#)**) to reduce bundle size. We provide two ways to customize:

## Custom component

The first way is to use `generatePicker` (or `generateCalendar`) to help create Picker components.

First, we initialize an antd demo with `create-react-app`. You can refer to [Use in TypeScript](#), or you can start directly here [init antd](#)

### DatePicker.tsx

Create `src/components/DatePicker.tsx`.

For example:

```
import { Dayjs } from 'dayjs';
import dayjsGenerateConfig from 'rc-picker/lib/generate/dayjs';
import generatePicker from 'antd/es/date-picker/generatePicker';
import 'antd/es/date-picker/style/index';

const DatePicker = generatePicker<Dayjs>(dayjsGenerateConfig);

export default DatePicker;
```

### TimePicker.tsx

Create `src/components/TimePicker.tsx`.

For example:

```
import { Dayjs } from 'dayjs';
import * as React from 'react';
import DatePicker from './DatePicker';
import { PickerTimeProps } from 'antd/es/date-picker/generatePicker';

export interface TimePickerProps extends Omit<PickerTimeProps<Dayjs>, 'picker'> {}

const TimePicker = React.forwardRef<any, TimePickerProps>((props, ref) => {
  return <DatePicker {...props} picker="time" mode={undefined} ref={ref} />;
});

TimePicker.displayName = 'TimePicker';

export default TimePicker;
```

### Calendar.tsx

Create `src/components/Calendar.tsx`.

For example:

```
import { Dayjs } from 'dayjs';
import dayjsGenerateConfig from 'rc-picker/lib/generate/dayjs';
import generateCalendar from 'antd/es/calendar/generateCalendar';
import 'antd/es/calendar/style';

const Calendar = generateCalendar<Dayjs>(dayjsGenerateConfig);

export default Calendar;
```

## Export Custom component

Create `src/components/index.tsx`.

For example:

```
export { default as DatePicker } from './DatePicker';
export { default as Calendar } from './Calendar';
export { default as TimePicker } from './TimePicker';
```

## Use Custom component

Modify `src/App.tsx`, import `dayjs` and custom component.

```
- import { DatePicker, Calendar } from 'antd';
- import format from 'moment';

+ import { DatePicker, TimePicker, Calendar } from './components';
+ import format from 'dayjs';
```

If the above steps do not work correctly, you can refer to [antd4-generate-picker/antd-ts](#).

If you need JavaScript code, you can refer to [antd4-generate-picker/antd-demo](#).

If you use [umi](#), you can reference [antd4-use-dayjs-replace-moment](#).

## antd-dayjs-webpack-plugin

We also provide another implementation, which we provide with `antd-dayjs-webpack-plugin`, replacing `momentjs` with `Day.js` directly without changing a line of existing code. More info can be found at [antd-dayjs-webpack-plugin](#).

```
// webpack-config.js
import AntdDayjsWebpackPlugin from 'antd-dayjs-webpack-plugin';

module.exports = {
  // ...
  plugins: [new AntdDayjsWebpackPlugin()],
};
```

## Use date-fns

[date-fns](#) currently supports custom component methods similar to `dayjs`. The difference is that the parameter types used are different. Support is provided in antd 4.5.0 and above.

For Example:

### DatePicker.tsx

Create `src/components/DatePicker.tsx`.

Code as follows:

```
import dateFnsGenerateConfig from 'rc-picker/lib/generate/dateFns';
import generatePicker from 'antd/es/date-picker/generatePicker';
import 'antd/es/date-picker/style/index';

const DatePicker = generatePicker<Date>(dateFnsGenerateConfig);

export default DatePicker;
```