

你可以用自定义日期库 ([day.js](#)、[date-fns](#)) 替换 Moment 以优化打包大小。在这里我们提供了两种方式来实现在替换:

自定义组件

第一种方法是使用 `generatePicker` (或 `generateCalendar`) 辅助创建 Picker 组件。

我们先初始化一个 `create-react-app` 的 antd demo, 你可以参考 [在 TypeScript 中使用](#) 进行构建, 也可以直接从[这里开始](#)[init antd](#)

DatePicker.tsx

新建 `src/components/DatePicker.tsx`。

编写如下代码:

```
import { Dayjs } from 'dayjs';
import dayjsGenerateConfig from 'rc-picker/es/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

新建 `src/components/TimePicker.tsx`。

编写如下代码:

```
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

新建 `src/components/Calendar.tsx`。

编写如下代码:

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

const Calendar = generateCalendar<Dayjs>(dayjsGenerateConfig);

export default Calendar;
```

导出自定义组件

新建 `src/components/index.tsx`。

编写如下代码:

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

使用自定义组件

修改 `src/App.tsx`，引入 `dayjs` 和自定义的组件。

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

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

如果按照上述步骤无法正确运行的话，你可以参考[antd4-generate-picker/antd-ts](https://github.com/ant-design/ant-design/issues/15444)。

如果你需要 JavaScript 代码，你可以参考 [antd4-generate-picker/antd-demo](https://github.com/ant-design/ant-design/issues/15444)。

如果你熟悉 [umi](https://umijs.org/)，你可以参考 [antd4-use-dayjs-replace-moment](https://github.com/ant-design/ant-design/issues/15444)。

antd-dayjs-webpack-plugin

我们还提供另一种实现方式。使用 `antd-dayjs-webpack-plugin` 插件，无需对现有代码做任何修改直接替换成 `Day.js`。请参考 [antd-dayjs-webpack-plugin](https://github.com/ant-design/ant-design/issues/15444)。

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

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

使用 date-fns

[date-fns](#) 目前支持和 dayjs 类似的自定义组件方法，区别在于使用的参数类型不同，在 antd 4.5.0 以上版本提供支持。

做一个简单的例子：

DatePicker.tsx

新建 `src/components/DatePicker.tsx` 。

编写如下代码：

```
import dateFnsGenerateConfig from 'rc-picker/es/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;
```