

Migration from @material-ui-pickers

@material-ui/pickers was moved to the @mui/lab.

⚠ The date picker components were rewritten. In most places, the logic was rewritten from scratch, so it isn't possible to maintain the whole list of changes. Here's an overview of the most important concepts that were changed. If you are going to upgrade, the easiest way might be to go through each picker usage in your codebase, and rewrite them one at a time. Don't forget to run your tests after each!

This guide is an overview of the core concepts that were changed from pickers v3.2.10.

Installation

You need to install the @mui/lab package if it's not already installed. ⚠ Make sure you have installed the latest version, "@mui/lab": ^5.0.0-alpha.30" or above.

Imports

The keyboard version of pickers is no longer published. All versions of mobile and desktop pickers implement keyboard input for accessibility.

```
-import { KeyboardDatePicker } from '@material-ui/pickers';
+import DatePicker from '@mui/lab/DatePicker';

-<KeyboardDatePicker />
+<DatePicker />
```

Also, instead of providing a variant prop, these were moved to different imports, meaning that your bundle won't include Dialog if you are using only the desktop picker.

- <DesktopDatePicker /> – Only desktop view.
- <MobileDatePicker /> – Only mobile view.
- <DatePicker /> – Mobile or Desktop view according to the user **pointer** preference.
- <StaticDatePicker /> – The picker view itself, without input or any other wrapper.

```
-import { DatePicker } from '@material-ui/pickers';
+import DesktopDatePicker from '@mui/lab/DesktopDatePicker';

-<DatePicker variant="inline" />
+<DesktopDatePicker />
```

The same convention applies to TimePicker – <DesktopTimePicker> and <MobileTimePicker /> .

MuiPickersUtilsProvider

The MuiPickersUtilsProvider was removed in favor of LocalizationProvider . Also, pickers do not require you to install date-io adapters manually. Everything is included with the lab .

✗ Before:

```
import AdapterDateFns from '@date-io/date-fns';
import { MuiPickersUtilsProvider } from '@material-ui/pickers';
```

✓ After:

```
import AdapterDateFns from '@mui/lab/AdapterDateFns';
import LocalizationProvider from '@mui/lab/LocalizationProvider';

function App() {
  return (
    <LocalizationProvider dateAdapter={AdapterDateFns}>
      ...
    </LocalizationProvider>
  )
};
```

Render input

We introduced a new **required** `renderInput` prop. This simplifies using non-MUI text field input components.

```
<DatePicker renderInput={(props) => <TextField {...props} />} />
<TimePicker renderInput={(props) => <TextField {...props} />} />
```

Previously, props were spread on the `<TextField />` component. From now on you will need to use the new `renderInput` prop to provide these:

```
<DatePicker
-  label="Date"
-  helperText="Something"
+  renderInput={props => <TextField label="Date" helperText="Something" /> }
/>
```

State management

The state/value management logic for pickers was rewritten from scratch. Pickers will now call the `onChange` prop when each view of the date picker ends is completed. The `onError` handler is also completely different. Triple-check your pickers with forms integration, because form-integration issues can be subtle.

No required mask

Mask is no longer required. Also, if your provided mask is not valid, pickers will just ignore the mask, and allow arbitrary input.

```
<DatePicker
  mask="mm"
  value={new Date()} />
```

```

    onChange={console.log}
    renderInput={ (props) => (
      <TextField {...props} helperText="invalid mask" />
    ) }
  />

  <DatePicker
    value={new Date()}
    onChange={console.log}
    renderInput={ (props) => (
      <TextField {...props} helperText="valid mask" />
    ) }
  />

```

And many more

- ```

 <DatePicker
 - format="DD-MMM-YYYY"
 + inputFormat="DD-MMM-YYYY"
 />

```

There are many changes, be careful, make sure your tests, and build pass. In the event you have an advanced usage of the date picker, it will likely be simpler to rewrite it.

Please open a pull request to improve the guide if you notice an opportunity for doing such.