



# Flutter Material Date Picker Redesign

## SUMMARY

Update the Material Date Picker to match the latest design.

**Author:** Darren Austin (@darrenaustin)

**Go Link:** [flutter.dev/go/material-date-picker-redesign](https://flutter.dev/go/material-date-picker-redesign)

**Created:** 03/2020 / **Last updated:** 03/2020

## OBJECTIVE

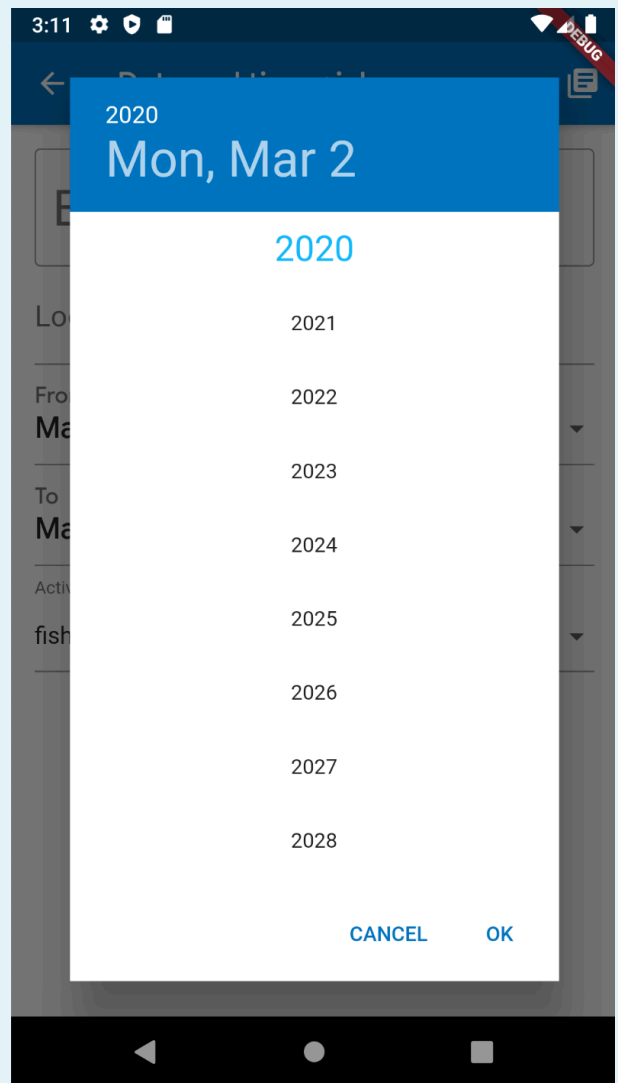
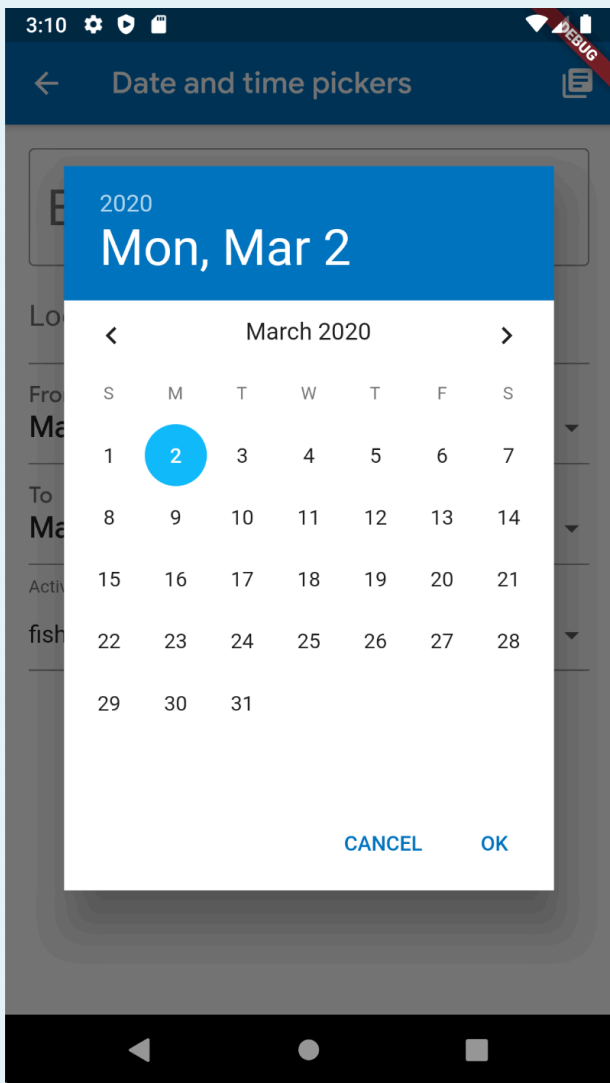
The design for the Material Design Date Picker has been updated:

<https://material.io/components/pickers/#mobile-pickers>

As the user interface for this component has changed substantially, we need to make changes to the public API to support the new design. This document describes these changes and includes a migration path for existing applications.

## BACKGROUND

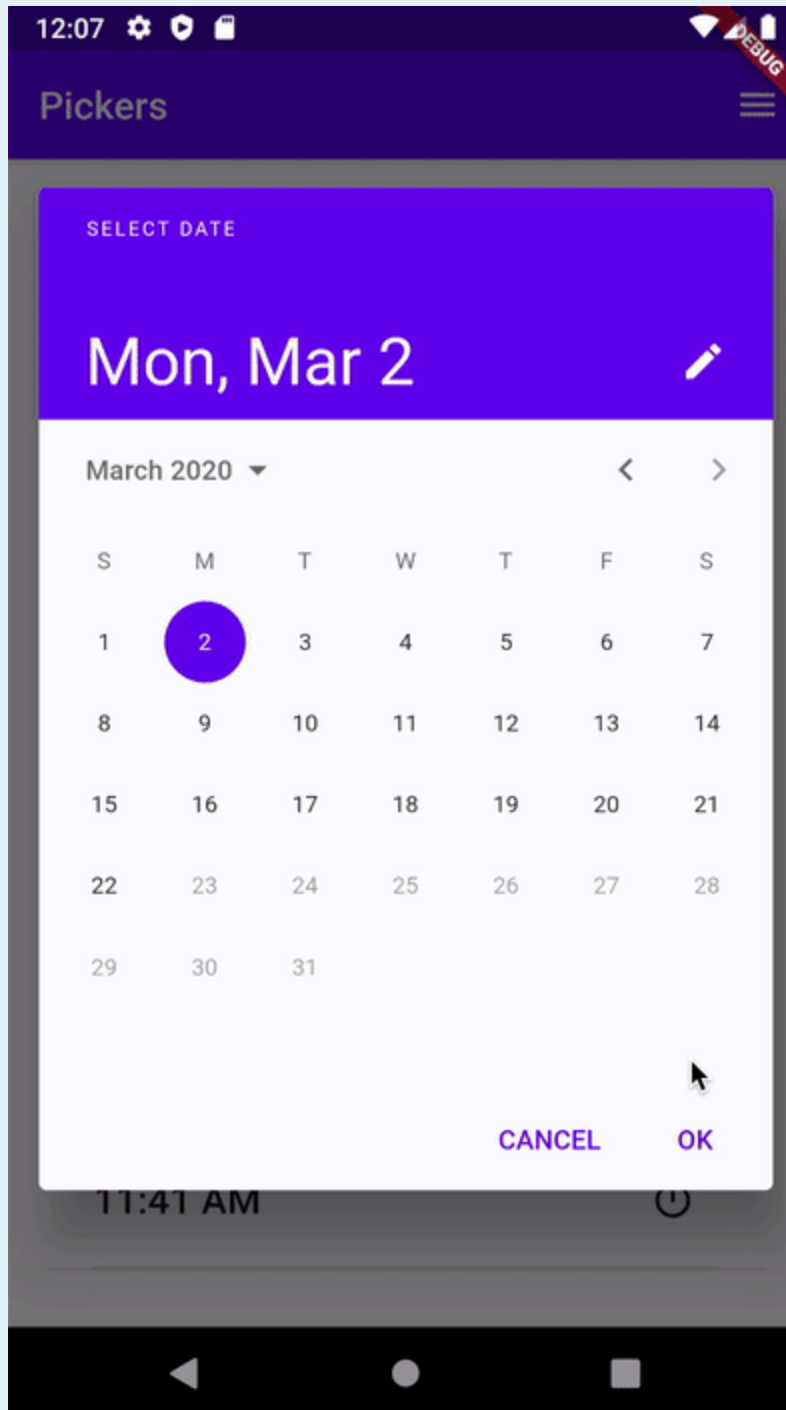
The current Flutter Material Date Picker is invoked via the `showDatePicker()` method. This presents the user with a dialog that allows them to pick a date:



The current Flutter Material Date Picker API consists of the following:

- **showDatePicker()** (top level function)
- **DayPicker** (widget showing a grid of days for a given month)
- **MonthPicker** (widget wrapping the **DayPicker** with a header that selects the a month to display)
- **YearPicker** (widget that displays a list of years)
- **DatePickerMode** (enum: day, year)
- **SelectableDayPredicate** (function typedef)

The new design has a different header and includes the ability to switch to a new text input field version of the dialog:



## OVERVIEW

In order to support the [new design](#), we are replacing the existing implementation and API with one that matches the new design. The existing **DayPicker**, **MonthPicker**, and **YearPicker** classes will be deprecated as they don't match the new UI design. We will be exposing new **CalendarDatePicker** and **InputDatePickerFormField** classes for applications that need to implement

custom date picker interfaces.

## Non-goals

This design doesn't include the new Date Range picker user interface. That will be coming in a separate PR as it doesn't include any breaking changes.

## DETAILED DESIGN/DISCUSSION

A proposed PR that implements this design update:

<https://github.com/flutter/flutter/pull/50546>

Apps that only relied only on `showDatePicker()` will not have to make any changes to use the new UI design. There are **new parameters** to this function to support the new text input mode and configure text fields that appear in the dialog:

```
Future<DateTime> showDatePicker({
  @required BuildContext context,
  @required DateTime initialDate,
  @required DateTime firstDate,
  @required DateTime lastDate,
  DatePickerEntryMode initialEntryMode = DatePickerEntryMode.calendar,
  SelectableDayPredicate selectableDayPredicate,
  String helpText,
  String cancelText,
  String confirmText,
  Locale locale,
  bool useRootNavigator = true,
  RouteSettings routeSettings,
  TextDirection textDirection,
  TransitionBuilder builder,
  DatePickerMode initialDatePickerMode = DatePickerMode.day,
  String errorFormatText,
  String errorInvalidText,
  String fieldHintText,
  String fieldLabelText,
})
```

The existing **DayPicker**, **MonthPicker**, and **YearPicker** classes are not very useful APIs as they exist today. There are even notes in the docs to discourage their use. Given that their implementation had to be changed and that no internal clients are using them directly, we are deprecating these classes. To replace them we will be exposing two new classes that are used to implement the new Date Picker:

```

CalendarDatePicker({
  Key key,
  @required DateTime initialDate,
  @required DateTime firstDate,
  @required DateTime lastDate,
  @required this.onDateChanged,
  this.onDisplayedMonthChanged,
  this.initialCalendarMode = DatePickerMode.day,
  this.selectableDayPredicate,
})

```

Which implements the calendar grid and year picker interface of the new date picker. This may be useful for embedding this picker in another dialog or on a sidebar for a calendar desktop app.

For adding apps that want to add a text input field to a form that supports dates, apps can use the new:

```

InputDatePickerFormField({
  Key key,
  DateTime initialDate,
  @required DateTime firstDate,
  @required DateTime lastDate,
  this.onDateSubmitted,
  this.onDateSaved,
  this.selectableDayPredicate,
  this.errorFormatText,
  this.errorInvalidText,
  this.fieldHintText,
  this.fieldLabelText,
  this.autofocus = false,
})

```

## OPEN QUESTIONS

- Should we expose the entire dialog as a **DatePickerDialog**? It is currently private, but we have had [requests](#) to open it up to the public.

## TESTING PLAN

The existing [date picker tests](#) have been updated to match the new user interface as well as adding tests for the new functionality.