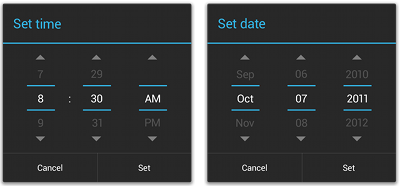
# Pickers

Android provides controls for the user to pick a time or pick a date as ready-to-use dialogs. Each picker provides controls for selecting each part of the time (hour, minute, AM/PM) or date (month, day, year). Using these pickers helps ensure that your users can pick a time or date that is valid, formatted correctly, and adjusted to the user's locale.



We recommend that you use [DialogFragment](https://developer.android.com/reference/androidx/fragment/app/DialogFragment) to host each time or date picker. The [DialogFragment](https://developer.android.com/reference/androidx/fragment/app/DialogFragment) manages the dialog lifecycle for you and allows you to display the pickers in different layout configurations, such as in a basic dialog on handsets or as an embedded part of the layout on large screens.

Although [DialogFragment](https://developer.android.com/reference/android/app/DialogFragment) was first added to the platform in Android 3.0 (API level 11), if your app supports versions of Android older than 3.0—even as low as Android 1.6—you can use the [DialogFragment](https://developer.android.com/reference/androidx/fragment/app/DialogFragment) class that's available in the [support library](https://developer.android.com/tools/support-library) for backward compatibility. Key classes are the following:

* [DatePickerDialog](https://developer.android.com/reference/android/app/DatePickerDialog)
* [TimePickerDialog](https://developer.android.com/reference/android/app/TimePickerDialog)

## Creating a Date Picker

Creating a [DatePickerDialog](https://developer.android.com/reference/android/app/DatePickerDialog) is just like creating a [TimePickerDialog](https://developer.android.com/reference/android/app/TimePickerDialog). The only difference is the dialog you create for the fragment.

To display a [DatePickerDialog](https://developer.android.com/reference/android/app/DatePickerDialog) using [DialogFragment](https://developer.android.com/reference/androidx/fragment/app/DialogFragment), you need to define a fragment class that extends [DialogFragment](https://developer.android.com/reference/androidx/fragment/app/DialogFragment) and return a [DatePickerDialog](https://developer.android.com/reference/android/app/DatePickerDialog) from the fragment's [onCreateDialog()](https://developer.android.com/reference/androidx/fragment/app/DialogFragment" \l "onCreateDialog(android.os.Bundle)) method.

### Extending DialogFragment for a date picker

To define a [DialogFragment](https://developer.android.com/reference/androidx/fragment/app/DialogFragment) for a [DatePickerDialog](https://developer.android.com/reference/android/app/DatePickerDialog), you must:

* Define the [onCreateDialog()](https://developer.android.com/reference/androidx/fragment/app/DialogFragment" \l "onCreateDialog(android.os.Bundle)) method to return an instance of [DatePickerDialog](https://developer.android.com/reference/android/app/DatePickerDialog)
* Implement the [DatePickerDialog.OnDateSetListener](https://developer.android.com/reference/android/app/DatePickerDialog.OnDateSetListener) interface to receive a callback when the user sets the date.
* public static class DatePickerFragment extends DialogFragment  
                              implements DatePickerDialog.OnDateSetListener {  
    
      @Override  
      public Dialog onCreateDialog(Bundle savedInstanceState) {  
          // Use the current date as the default date in the picker  
          final Calendar c = Calendar.getInstance();  
          int year = c.get(Calendar.YEAR);  
          int month = c.get(Calendar.MONTH);  
          int day = c.get(Calendar.DAY\_OF\_MONTH);  
    
          // Create a new instance of DatePickerDialog and return it  
          return new DatePickerDialog(getActivity(), this, year, month, day);  
      }  
    
      public void onDateSet(DatePicker view, int year, int month, int day) {  
          // Do something with the date chosen by the user  
      }  
  }
* See the [DatePickerDialog](https://developer.android.com/reference/android/app/DatePickerDialog) class for information about the constructor arguments.
* Now all you need is an event that adds an instance of this fragment to your activity.

### Showing the date picker

* Once you've defined a [DialogFragment](https://developer.android.com/reference/androidx/fragment/app/DialogFragment) like the one shown above, you can display the date picker by creating an instance of the [DialogFragment](https://developer.android.com/reference/androidx/fragment/app/DialogFragment) and calling [show()](https://developer.android.com/reference/androidx/fragment/app/DialogFragment#show(android.support.v4.app.FragmentManager,%20java.lang.String)).
* For example, here's a button that, when clicked, calls a method to show the dialog:
* <Button  
      android:layout\_width="wrap\_content"  
      android:layout\_height="wrap\_content"  
      android:text="@string/pick\_date"  
      android:onClick="showDatePickerDialog" />
* When the user clicks this button, the system calls the following method:
* public void showDatePickerDialog(View v) {  
      DialogFragment newFragment = new DatePickerFragment();  
      newFragment.show(getSupportFragmentManager(), "datePicker");  
  }  
  This method calls [show()](https://developer.android.com/reference/androidx/fragment/app/DialogFragment#show(android.support.v4.app.FragmentManager,%20java.lang.String)) on a new instance of the [DialogFragment](https://developer.android.com/reference/androidx/fragment/app/DialogFragment) defined above. The [show()](https://developer.android.com/reference/androidx/fragment/app/DialogFragment#show(android.support.v4.app.FragmentManager,%20java.lang.String)) method requires an instance of [FragmentManager](https://developer.android.com/reference/androidx/fragment/app/FragmentManager) and a unique tag name for the fragment.

**<?xml** version="1.0" encoding="utf-8"**?>**

**<RelativeLayout** xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    tools:context="example.javatpoint.com.datepicker.MainActivity"**>**

**<TextView**

        android:id="@+id/textView1"

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:layout\_above="@+id/button1"

        android:layout\_alignParentLeft="true"

        android:layout\_alignParentStart="true"

        android:layout\_marginBottom="102dp"

        android:layout\_marginLeft="30dp"

        android:layout\_marginStart="30dp"

        android:text="" **/>**

**<Button**

        android:id="@+id/button1"

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:layout\_alignParentBottom="true"

        android:layout\_centerHorizontal="true"

        android:layout\_marginBottom="20dp"

        android:text="Change Date" **/>**

**<DatePicker**

        android:id="@+id/datePicker"

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:layout\_above="@+id/textView1"

        android:layout\_centerHorizontal="true"

        android:layout\_marginBottom="36dp" **/>**

**</RelativeLayout>**

*MainActivity.java*

**package** example.javatpoint.com.datepicker;

**import** android.support.v7.app.AppCompatActivity;

**import** android.os.Bundle;

**import** android.view.View;

**import** android.widget.Button;

**import** android.widget.DatePicker;

**import** android.widget.TextView;

**public** **class** MainActivity **extends** AppCompatActivity {

    DatePicker picker;

    Button displayDate;

    TextView textview1;

    @Override

**protected** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_main);

        textview1=(TextView)findViewById(R.id.textView1);

        picker=(DatePicker)findViewById(R.id.datePicker);

        displayDate=(Button)findViewById(R.id.button1);

        textview1.setText("Current Date: "+getCurrentDate());

        displayDate.setOnClickListener(**new** View.OnClickListener(){

            @Override

**public** **void** onClick(View view) {

                textview1.setText("Change Date: "+getCurrentDate());

            }

        });

    }

**public** String getCurrentDate(){

        StringBuilder builder=**new** StringBuilder();;

        builder.append((picker.getMonth() + 1)+"/");//month is 0 based

        builder.append(picker.getDayOfMonth()+"/");

        builder.append(picker.getYear());

**return** builder.toString();

    }

}

## Creating a Time Picker

To display a [TimePickerDialog](https://developer.android.com/reference/android/app/TimePickerDialog) using [DialogFragment](https://developer.android.com/reference/androidx/fragment/app/DialogFragment), you need to define a fragment class that extends [DialogFragment](https://developer.android.com/reference/androidx/fragment/app/DialogFragment) and return a [TimePickerDialog](https://developer.android.com/reference/android/app/TimePickerDialog) from the fragment's [onCreateDialog()](https://developer.android.com/reference/androidx/fragment/app/DialogFragment" \l "onCreateDialog(android.os.Bundle)) method.

**Note:** If your app supports versions of Android older than 3.0, be sure you've set up your Android project with the support library as described in [Setting Up a Project to Use a Library](https://developer.android.com/tools/support-library/setup).

### Extending DialogFragment for a time picker

To define a [DialogFragment](https://developer.android.com/reference/androidx/fragment/app/DialogFragment) for a [TimePickerDialog](https://developer.android.com/reference/android/app/TimePickerDialog), you must:

* Define the [onCreateDialog()](https://developer.android.com/reference/androidx/fragment/app/DialogFragment" \l "onCreateDialog(android.os.Bundle)) method to return an instance of [TimePickerDialog](https://developer.android.com/reference/android/app/TimePickerDialog)
* Implement the [TimePickerDialog.OnTimeSetListener](https://developer.android.com/reference/android/app/TimePickerDialog.OnTimeSetListener) interface to receive a callback when the user sets the time.

Here's an example:

public static class TimePickerFragment extends DialogFragment  
                            implements TimePickerDialog.OnTimeSetListener {  
  
    @Override  
    public Dialog onCreateDialog(Bundle savedInstanceState) {  
        // Use the current time as the default values for the picker  
        final Calendar c = Calendar.getInstance();  
        int hour = c.get(Calendar.HOUR\_OF\_DAY);  
        int minute = c.get(Calendar.MINUTE);  
  
        // Create a new instance of TimePickerDialog and return it  
        return new TimePickerDialog(getActivity(), this, hour, minute,  
                DateFormat.is24HourFormat(getActivity()));  
    }  
  
    public void onTimeSet(TimePicker view, int hourOfDay, int minute) {  
        // Do something with the time chosen by the user  
    }  
}

### Showing the time picker

Once you've defined a [DialogFragment](https://developer.android.com/reference/androidx/fragment/app/DialogFragment) like the one shown above, you can display the time picker by creating an instance of the [DialogFragment](https://developer.android.com/reference/androidx/fragment/app/DialogFragment) and calling [show()](https://developer.android.com/reference/androidx/fragment/app/DialogFragment#show(android.support.v4.app.FragmentManager,%20java.lang.String)).

For example, here's a button that, when clicked, calls a method to show the dialog:

<Button  
    android:layout\_width="wrap\_content"  
    android:layout\_height="wrap\_content"  
    android:text="@string/pick\_time"  
    android:onClick="showTimePickerDialog" />

When the user clicks this button, the system calls the following method

public void showTimePickerDialog(View v) {  
    DialogFragment newFragment = new TimePickerFragment();  
    newFragment.show(getSupportFragmentManager(), "timePicker");  
}

This method calls [show()](https://developer.android.com/reference/androidx/fragment/app/DialogFragment#show(android.support.v4.app.FragmentManager,%20java.lang.String)) on a new instance of the [DialogFragment](https://developer.android.com/reference/androidx/fragment/app/DialogFragment) defined above. The [show()](https://developer.android.com/reference/androidx/fragment/app/DialogFragment#show(android.support.v4.app.FragmentManager,%20java.lang.String)) method requires an instance of [FragmentManager](https://developer.android.com/reference/androidx/fragment/app/FragmentManager) and a unique tag name for the fragment.

**Caution:** If your app supports versions of Android lower than 3.0, be sure that you call **[getSupportFragmentManager()](https://developer.android.com/reference/androidx/fragment/app/FragmentActivity" \l "getSupportFragmentManager())** to acquire an instance of **[FragmentManager](https://developer.android.com/reference/androidx/fragment/app/FragmentManager)**. Also make sure that your activity that displays the time picker extends **[FragmentActivity](https://developer.android.com/reference/androidx/fragment/app/FragmentActivity)** instead of the standard [**Activity**](https://developer.android.com/reference/android/app/Activity) class.

<TimePicker android:id="@+id/timePicker1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:timePickerMode="clock" />

<TimePicker  
 android:id="@+id/timePicker1"  
 android:layout\_width="357dp"  
 android:layout\_height="356dp"  
 android:layout\_above="@+id/set\_button"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="196dp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 tools:ignore="UnknownId" />